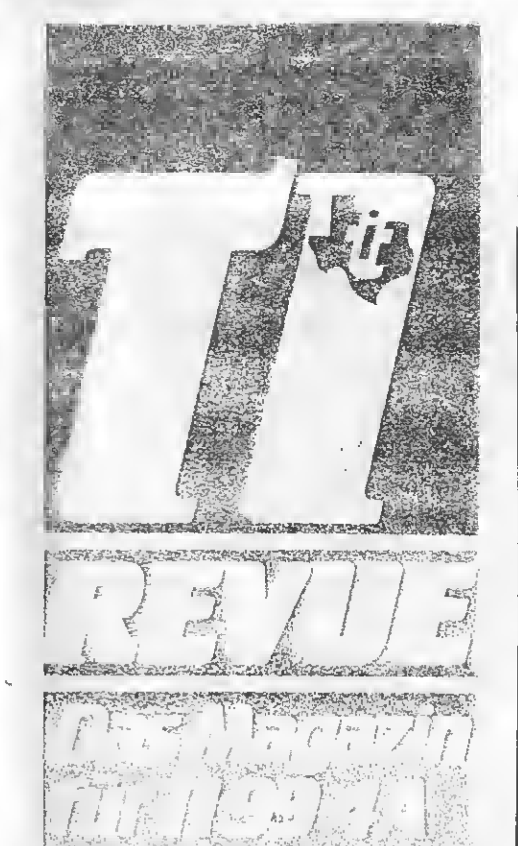


NR. 9/86 SEPTEMBER -DM 5,50 / ÖS 46 / SRF 5,50

BAUEN SIE SICH DOCH IHREN PLOTTER SELBST (III)

```
0140G  9C02  0788  2547  6F1E  5245  4744  592D  5052        READY-PR
0150G  4553  5320  414E  5920  4B45  5920  544F  2042   ESS ANY KEY TO B
0160G  4547  494E  F831  0011  A126  0452  3100  1802   EGIN 1   (   1
0170G  C404  8B31  000D  A1EA  04A3  BE43  1006  0379      1  }   C   y
0180G  87B0  D086  556E  6D04  0F19  6188  BF72  0080      U m    a  r
0190G  BF90  7201  9F0F  1ABD  9073  9072  9672  008F     r      s r  r
01A0G  80D0  418C  3900  0101  044D  8674  02FF  0341     A 9    M t   A
01B0G  AC06  03CB  0720  BE72  FEBE  6005  866C  6680          r  m  l
01C0G  FB31  001E  A420  6000  BE80  FB04  3100  1EA4    1             1
01D0G  2060  0086  5866  5905  01DC  9059  CE59  1061       X Y    Y Y a
01E0G  EDD4  E400  5BE4  2056  41EF  0501  DA41  FD94       X  X A     A
01F0G  72B7  9072  9472  BF92  7212  A190  6CD6  8FDD   r  r r    r    l
0200G  00AA  4224  BD58  8FDD  068F  5862  2494  72BF     B$ X     Xb$ r
0210G  9072  FFFF  9472  BD50  7258  906C  BD58  CF7D    r    r  rX l X )
0220G  0058  4209  9472  8790  720F  1A62  2496  72D7    XB  r   r  b$ r
0230G  0212  A142  4031  0001  5960  00D5  59AA  41B6      B@1  Y    Y A
0240G  3900  0101  044E  08A2  8102  0102  039C  0204   9     N
0250G  0506  9C02  0708  09A2  9E04  5052  4553  53FB              PRESS
0260G  3100  11A0  2804  9231  000D  A068  04A3  BF52   1   (   1    h   R
0270G  00E4  BE58  308E  6C42  9008  FF02  0F49  4E53     X0 lB        INS
0280G  4552  5420  4341  5254  5249  4447  45FB  42EC   ERT CARTRIDGE B
0290G  9058  BCB0  5258  9552  3100  0380  5209  49A3    X  RX R1    R I
02A0G  5200  04BD  6A90  7296  72BD  5C90  7296  72A3   R    j r r \ r r
02B0G  6A00  04B6  5E6E  5C62  CD35  0001  5FCF  7D00   j     ^ \b 5     )
02C0G  6A91  6A34  5EB0  52CF  7D00  6A42  DD39  0001   j j4^ R } jB 3
02D0G  5F00  006A  916A  325E  9052  0000  6AA3  5200   _  j j2^ R   j R
02E0G  3AD2  7200  6250  BE43  1306  0379  3900  0101   : r b  C    y9
02F0G  044D  B674  02FF  0342  F4A6  7531  C875  6C43    M l    B  u1 ulC
0300G  0706  03D3  0502  F406  03CB  A46C  7592  6CE2                lu l
0310G  6C02  BC78  906L  946C  BD5C  906C  955C  BF72   l   x l l \ l \ r
0320G  9E80  8E78  632F  BD82  B0CF  7000  5C43  3733       xc/      } \C73
0330G  0002  8080  0000  5C27  208F  60CE  4339  CF70            \      C9 p
0340G  1000  4353  BD00  70A7  006r  FF34  00AF  1000     CS   p     4
0350G  AF0F  FF66  0035  006F  0100  3500  3C80  8400        5  o  5 <
0360G  8674  3500  1FA3  81A3  8067  80B2  8E78  6376    t5                xcx
0370G  9673  BD00  8080  0FF0  00BE  4260  3100  0228    s           B'1   (
0380G  6000  D628  AA43  92D2  2900  6392  9473  BD50      ( C    ) c    s
0390G  7342  0031  0200  B04A  04B0  00BF  80D0  05B0   ≤B 1    J
03A0G  BE80  D24B  6650  4A33  0027  E001  4A00  02D0      e   JG        J
03B0G  A34A  0008  A380  D000  0792  80D2  43A4  009F     J               C
03C0G  80D6  0870  5E60  D21F  0583  A4BF  5804  75F6       p             X u
03D0G  5800  00BF  5504  8043  CF88  6D86  548C  55B0   X    X  C    m T  U
```

MIT DEM TI-STEUERN UND REGELN

DIREKTE JOYSTICK-ABFRAGE

ASSEMBLER DIALOG

NEUER COMPILER

TIPS & TRICKS

IMPRESSUM

TI-REVUE, die Zeitschrift für den TI PC und TI 99/4A, erscheint monatlich in der München Aktuell Verlags GmbH
Redaktion: Senator-Presse-Service.
Verantwortlich für den Inhalt: Heiner Martin.
Verantwortlich für Listings: Hartmut König.
Geschäftsführer: Werner E. Seibt
Alle: Postfach 1107, 8044 Lohhof. Anfragen bitte nur schriftlich.
Druck: Maier und Söhne
Es gilt die Honorarliste des Verlages. Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung.
Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck sowie die Aufnahme in den Programm-Service nach den Verlags-Sätzen und überträgt dem Verlag das Copyright!
Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jedwede Verwertung ist untersagt. Nachdruck nur mit ausdrücklicher schriftlicher Zustimmung des Verlages.
Namentlich gezeichnete Artikel geben nicht unbedingt die Meinung der Redaktion wieder.
Kein Anspruch auf Lieferung bei Ausfall durch höhere Gewalt.
Abo- und Kassetten-Service: Henny Rose Seibt
©by TI/CBM Verlag SPS und Autoren.
★
Gerichtsstand: München

Grüß Gott - Guten Tag

Die im Computerhobby immer etwas ruhigere Sommerzeit ist nun fast vorbei und es geht wieder munter mit der Computerei weiter. Dieser Herbst wird nach unseren Informationen noch einmal eine ganze Menge Neuheiten für den TI 99/4A bringen. Wer hätte gedacht, daß der TI 99/4A so ein langes Leben nach seiner Produktionseinstellung haben würde. Und ein Ende ist eigentlich noch nicht in Sicht.

Für den Herbst dürfen wir wohl die Einführung der 80-Zeichen-Erweiterung erwarten. Aus den USA und auch aus Deutschland haben wir von neuen Karten für die P-Box gehört, dazu gehört auch die digitale Ein-/Ausgabe. Neueste Gerüchte aus den USA sprechen auch davon, daß neben der Fa. Myarc noch ein Konsortium aus mehreren Herstellern an einem neuen Computer arbeiten. Nach unseren Informationen hat dieses Konsortium einen deutschen Programmierer um Mithilfe gebeten. So hoffen wir, hier aus erster Hand bald weitere Informationen zu erhalten.

Unabhängig davon aber ist unser TI 99/4A wohl doch auch weiterhin eine interessante Maschine. Einer unserer Redakteure war jetzt aus beruflichen Gründen gezwungen, tiefer in die Welt des Personal Computers einzudringen, und wenn man ihn so reden hört, scheint er nicht sehr begeistert zu sein. Das DOS (wenn auch etwas langsam, so doch fehlerfrei), das Betriebssystem und auch den Prozessor des TI 99/4A hält er immer noch für aktuell. So sagt er unter anderem, daß sein TI 99/4A auf gar keinen Fall wegen des PC arbeitslos wird.

Die Aussichten für den TI 99/4A stehen damit auch in Zukunft nicht schlecht. Wir hoffen, in der nächsten Ausgabe von weiteren Neuheiten berichten zu können. Bis dahin die besten Grüße

von Eurem

TI-REVUE-Team

Nicht vergessen: Seid Ihr mit dem Heft zufrieden, sagt es weiter, seid Ihr unzufrieden, sagt es uns.

Und: Jeden Dienstag von 15 bis 19 Uhr stehen Ihnen unsere Lesertelefone zur Verfügung. Für technische und Assembler-Fragen Tel.-Nr. 0731/33220 und zu den Listings bzw. Fragen zu den Abonnements und dem Kassettenservice Tel.-Nr. 089/1298013

Wichtiger Hinweis des Verlages!

Aufgrund verschiedener Umstände sieht sich der Verlag gezwungen, ab dieser Ausgabe den Versand von Programmkassetten einzustellen. Für zurückliegende Hefte noch vorhandene Kassetten werden selbstverständlich weiter ausgeliefert.

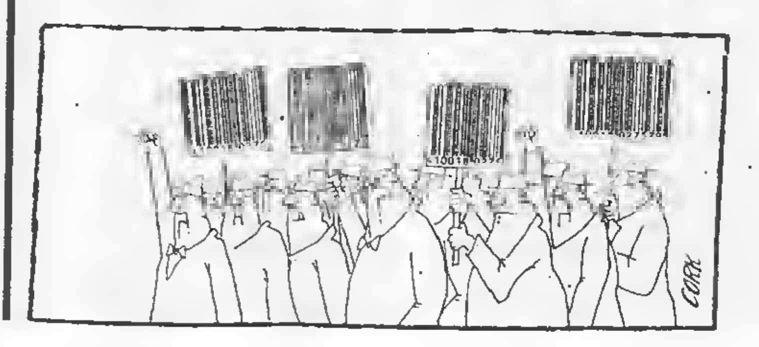
Ab sofort – beginnend mit der Ausgabe 9/86 der TI-REVUE – werden die in der TI-REVUE veröffentlichten Listings nur noch auf Diskette abgegeben!

INHALT 9/86

DIALOG

REPORT & SERVICE

LISTINGS



DIALOG

TASTATUR PRELLT – KEIN GRUND ZUR AUFGABE

Ich finde es schade, daß viele TI-User das Handtuch werfen, nur weil Ihre Tastatur prellt. Für nur 18,90 DM kann man hierbei Abhilfe schaffen. Der Elektronikversand ‚Bühler Elektronik' führt in seinem Angebot eine Tastatur, die wahrscheinlich für den in den USA erhältlich gewesenen beigen TI-99 bestimmt war. Der Unterschied zu der herkömmlichen Tastatur besteht darin, daß erstens die Tasten besser und dadurch stabiler gekapselt wurden. Der zweite Unterschied besteht in der Farbe. Die Tastenkappen sind beige mit dunkler Schrift. Diese Farbkombination sieht jedoch beim normalen TI durchaus nicht schlecht aus. Aufgrund dieses Angebotes sollte sich so mancher TI-User überlegen, ob er sich ein neues System oder eine neue Tastatur zulegt.
Peter Kliem,
Erftstadt

Vielen Dank für den nützlichen Hinweis. Durch diese neue Tastatur werden hoffentlich einige TI-Benützer zu ihrem Computer zurückfinden. Letzte Meldung: Schlechte Nachricht, die Tastaturen sind ausverkauft.

SINGLE STEP FEHLERHAFT

Um den E/A besser nutzen zu können, habe ich mir den Single Step aus der TI-REVUE 3/84 zusammengebaut. Anschließend verdrahtete ich diese mit der Platine aus TI-REVUE 6/85, anhand dieser man den Single Step in der P-Box betreiben kann. Leider reagierte der so neu entstandene Single Step nicht in der gewünschten Weise. Er stoppt zwar bei einem Breakpoint, jedoch führt er das Programm mit dem Befehl 'S' weiter, anstatt es im Single Step abzuarbeiten. In meinem Montageplan und auf der Platine konnten weder ich noch eine weitere Person einen Fehler finden. Die einzige Abweichung vom Schaltungsvorschlag stellen die von mir verwendeten 'LS-' statt 'ALS-' IC's dar. Was muß ich tun, um die Schaltung so zu prüfen, daß ich sie anschließend zum Laufen bringe? Bisher ging ich immer folgendermaßen vor:

- Laden eines Testprogramms in den E/A
- Laden und starten des Debuggers
- Setzen eines Breakpoints im Testprogramm
- Verlassen des Debuggers durch 'QUIT'
- Starten des Testprogrammes

Der Debugger stoppt, wie schon erwähnt, an der Stelle, an der sich der Breakpoint befindet. Es ist jedoch unmöglich, das Programm im Single Step abzuarbeiten. Der Debugger führt dann das Programm weiter und stoppt erst wieder am nächsten Breakpoint. Dieser ist zwar dann abgearbeitet, jedoch steht auf der Breakpointadresse der 'XOP'-Befehl.
Jeder weitere Versuch, das Programm anhand des 'S'-Befehls „weiterzuschalten", scheitert.
Eine Messung am Pin 15 des 74LS138 zeigt zu diesem Zeitpunkt einen negativen Nadelimpuls. Weiterhin ist es mir noch nicht klar, ob die Schaltung für die optische Anzeige fehlerfrei ist, da die Diode während des gesamten Testbetriebes leuchtete.
Während meiner Tests sind meine drei Karten (32k, Flex, Disk) kaputtgegangen. Während des Entfernens des I/O-Stekkers vergaß ich, die P-Box auszuschalten. Besteht die Möglichkeit, daß bei allen drei Karten die selben IC's defekt sind, wenn ja, welche?
Rainer Kierdorf,
Hamburg

Zuerst kommen wir einmal zu den defekten Karten. Es besteht durchaus die Möglichkeit, daß bei allen dreien der selbe Defekt aufgetreten ist. Da ein solcher Defekt schwer zu finden ist, raten wir Ihnen, zuerst bei der Flex-Karte alle IC's systematisch auszuwechseln. Sind Sie dann bei dem/den richtigen angekommen, so verfolgen Sie die Leitungen zu den anderen Karten. Tauschen Sie dann dort den/die entsprechenden Treiber IC's (LS244 und LS245) aus. Zu Ihrer Single-Step-Schaltung können wir Ihnen nur raten, die Schaltung zuerst einmal am seitlichen Port zu überprüfen. Treten dort auch die selben Fehler auf, so liegt entweder ein Bauteilfehler oder eine falsche Verbindung vor.

WELCHES KONTAKTSPRAY IST GUT FÜR DEN MODULPORT?

Bei mir haben sich wieder einige Fragen über den TI angestaut, die ich gerne loswerden möchte.
Wozu dient der zweite Kassettenrekorder 'CS2' beim TI?
Ich habe, wie viele andere TI-User wohl auch, Probleme mit den Kontakten des Modulports. Um eine geeignete Reinigung durchzuführen, legte ich mir das Kontaktspray 'Kontakt 60' zu. Können Sie mir sagen, ob dieses Spray für die Reinigung des Modulports geeignet ist?
Welche Bücher können Sie mir empfehlen, die mir, am besten deutsch, den Einstieg in die Maschinensprache zeigen?
Meine Gerätschaft besteht aus EX-BASIC, 32k und BASIC-Assembler.
Reinhard Kern,
Afing, Österreich

Mit dem zweiten Kassettenrecorder haben Sie die Möglichkeit, zwei Kassettendateien gleichzeitig offen zu haben. Während Sie von CS1 Daten lesen, können Sie auf CS2 Daten schreiben. Umgekehrt ist dies nicht möglich, da man auf CS2 nur schreiben kann.
Das Kontaktspray 'Kontakt 60' ist insofern nicht geeignet, da es ölhaltig ist. Bitte verwenden Sie zur Reinigung der Kontakte am besten 'Kontakt WL'.
Assemblerbücher für den TI sind momentan nur sehr wenige auf dem Markt. Zum Einstieg in die Maschinensprache würden wir Ihnen den Assemblerkurs der Fa. Hagera empfehlen.

WAS IST MIT DER 80-ZEICHEN-KARTE?

Ich habe einige Fragen zum TI -99/4A, mit der Hoffnung, Sie können sie mir beantworten.
1.) Gibt es eine deutsche Übersetzung der Anleitung für die UCSD-Pascal-Karte, oder zumindest Literatur, die sich auf diese Karte bezieht?
2.) Was ist mit der 80-Zeichen-Karte und wo kann man diese Karte oder einen deutschen Nachbau kaufen?
3.) Kann man den Commodore Farbmonitor 1901 mit dem TI RGB-Modulator benutzen und wie läuft das dann mit der 80-Zeichen-Karte?
4.) wie spreche ich 'PIO' in UCSD-Pascal an?
Immo Hartmann,
Berlin

Wir werden versuchen, Ihre Fragen der Reihe nach zu beantworten.
zu 1.) Eine deutsche Übersetzung existiert unserer Meinung nach nicht. Es gibt jedoch eine ganze Reihe guter Literatur über UCSD-Pascal, die auch für den TI passen. UCSD-Pascal ist auf allen Computern nahezu gleich, so

daß fast alle Bücher über dieses Pascal verwendbar sind.
zu 2.) Die 80-Zeichen-Karte ist eine deutsche Entwicklung. Es wird jedoch noch einige Zeit verstreichen, bis diese auf den Markt gelangt.
zu 3.) Der RGB-Modulator besitzt einen analogen RGB-Ausgang. Der Anschluß an den Commodore Monitor wird, insofern dieser einen solchen besitzt, durchaus möglich sein.
zu 3.) Hinsichtlich der Ansteuerung von 'PIO' aus Pascal sind wir überfragt. Vielleicht findet sich ein Leser, der zu dieser Frage die Antwort weiß.

UNKONTROLLIERTE SYSTEMABSTÜRZE

Ich habe folgende Probleme: Trotz der Anwendung eines Reinigungssprays am Modulport treten folgende rätselhafte Erscheinungen auf: Es treten in völlig korrekten Zeilen Fehler auf, Zeilennummern werden umgeschmissen usw. Ist mein TI defekt? Diese Probleme traten erst nach dem Einbau der 32k-Erweiterung auf. Vorher stürzte er ‚normal' ab. Nach der Reinigung war für einige Tage Ruhe, jedoch traten danach die Probleme wieder auf.
Gibt es eine Möglichkeit, anhand eines E/A-Programmes die Cassettenrecordersteuerung anzusprechen, wenn ja, wie? Was für eine Funktion hat die Speicheradresse −31804? Nach jedem Call Load, das auf diese Adresse zeigt, bewegt sich der Cursor weiter, ohne daß eine Eingabe erfolgte.
Michael Kugelmann, Niederhausen

Die Abstürze kommen nicht nur von den Kontakten, in die die Module angesteckt werden, sondern auch von dem Stekker auf der Hauptplatine. In diesen Stecker wird der Modulsteckerteil eingesteckt. Ebenfalls können diese Abstürze auch von einem Defekt in der Speichererweiterung kommen. Kontrollieren Sie nocheinmal alle Kontakte auf mögliche Kurzschlüsse. Der Kassettenrecorder wird von der CRU-Adresse >002C gesteuert. Die Zustände der Steuerung können Sie dann durch 'SBO' und 'SBZ' umändern. Auf der Adresse −31804 ist der Userdefined Interrupt. Wenn Sie auf diese Adresse einen Wert legen, bringen Sie das Betriebssystem durcheinander. Der Computer arbeitet dann nämlich ca. alle 1/50 Sekunde das Maschinenprogramm ab, das auf dieser Adresse steht.

BEI DOUBLE SIDE 1440 SEKTOREN PRO SEITE

Da ich beim Kauf meiner Diskettenlaufwerke keine Gebrauchsanweisung mitgeliefert bekam, habe ich folgende Fragen zur Arbeitsweise dieser Laufwerke.
Ich betreibe diese Laufwerke mit meinem Corcomp Diskkontroller. Dieser wurde auf 40 Track/Side, Double Side, Double Density, eingestellt. So kann ich auf einer Diskette 1440 Sektoren initialisieren. Kann ich diese entsprechend mit Programmen belegen?
Sind bei einer solchen Diskette auf einer Seite 720 Sektoren untergebracht oder sind alle 1440 Sektoren auf einer Seite?
Wenn alle Sektoren auf einer Seite sind, kann ich dann mit einer Diskette, die zwei Kerben besitzt, die doppelte Menge an Sektoren belegen?
Leider wurde darüber in der TI-REVUE noch nichts geschrieben.
Dann habe ich noch eine Bitte: Ich möchte meinen Drucker, einen Seikosha GP 500 CPC, an die RS232-Schnittstelle anschließen. Es kommt mir jedoch so vor, als ob dazu zu wenig Kabel vorhanden wären.
Günter Bischoff, Stuttgart

Bei Ihren Laufwerken beschreiben Sie mit dem Corcomp Controller die Disketten doppelseitig, d.h., auf jeder Seite befinden sich 720 Sektoren. Diese 1440 Sektoren stehen Ihnen dann vollkommen zur Datenspeicherung zur Verfügung.
Ihr Drucker verfügt wahrscheinlich über keine RS232, sondern über eine Centronics-Schnittstelle. Ist dies der Fall, so müssen Sie den Drucker mit der PIO an der RS232-Karte verbinden.

MONITORANSCHLUSS AN DEN TI

Ich besitze einen Farbmonitor 'Sanyo CD 3220N' und möchte diesen an meinen TI 99/4A anschließen. Leider verfügt dieser Monitor nicht über einen Scart-Stecker, sondern ist mit einem achtpoligen Stecker versehen. Da mir das Ausgangssignal aus der Konsole nicht bekannt ist, bitte ich Sie um Ihre Hilfe:
Gibt es einen geeigneten RGB-Modulator für den TI-99/4A zu kaufen, und wenn ja, wo?
Haben Sie ein Schaltbild, mit dem man sich selbst einen solchen Modulator bauen kann?
Wolfgang Stolz, Hamburg

Ihr Monitor der Firma Sanyo besitzt eine RGB-TTL Eingang. Dies ist beim TI nicht gebräuchlich. So sind auch die von den verschiedenen Händlern angebotenen RGB-Modulatoren nicht dafür geeignet. Ein RGB-TTL-Signal ist ebenfalls anders aufbereitet als das mehr verbreitete RGB-Analog-Signal. Leider können wir Ihnen mit einem Schaltbild für einen passenden Modulator nicht dienen.

DEN RICHTIGEN MONITOR FÜR DEN TI

Ich möchte mir für meinen TI einen monochromen Monitor zulegen. Da ich nicht weiß, welche Signale am TI herausgeführt werden, ist mir völlig unklar, wie man ein solches Gerät anschließt. Ist ein einfaches Kabel ausreichend, oder kommt man um zusätzliche Elektronik nicht herum?
Der TI produziert in der Regel dunkle Schrift auf hellem Hintergrund.
Läßt sich dies hardwaremäßig umgehen?
Thomas Kneisel, Kleinostheim

Zum Anschluß eines normalen, einfarbigen Monitors benötigen Sie nur ein Kabel: Auf Pin 2 der Ausgangsbuchse am Computer zum Modulator liegt das Y-Signal, auf Pin 6 Masse ('GND'). Die Farbdarstellung können Sie nur softwaremäßig umstellen. So störend wirkt sich dies jedoch nicht auf dem Monitor aus.

VIDEODAT AM TI-99/4A

Da einige Zeit nach dem Einschalten der gesamte Bildschirm zitterte, durchlief und schwarz verschmierte, habe ich die Video-Endstufe untersucht. Die Verursacher waren die Transistoren 311/3906 und 24G/2222. Diese sind aber im Handel nicht zu erhalten. Wie heißen die handelsüblichen Bezeichnungen oder wo sind diese Transistoren sonst erhältlich?
Ich möchte gerne 'Videodat' (WDR-Computer-Club) am 'MBI-TI-99/4A auf Centronics' anschließen. Die MBI-Beschreibung gibt leider keinen Aufschluß darüber, ob es nur für Output oder auch zum Input verwendbar ist.

In der Videodat-Beschreibung steht, eine RS232-Schnittstelle sei , und das ist das MBI doch schließlich. Der Computer läßt jedoch keine Buchstaben auf dem Bildschirm erscheinen, sondern meldet sich immer mit 'I/O ERROR IN
Hans-Jürgen Püttmann, Oberhausen

Die handelsüblichen Bezeichnungen für die Transistoren lauten 2N3906 und 2N2222. Unter dieser Bezeichnung sind sie bestimmt in jedem Elektronikshop erhältlich. Beim MBI kommt es darauf an, welche Ausführung Sie davon besitzen. Das MBI-Centronics-Interface benutzt zunächst auch eine serielle Datenübertragung, bevor es auf parallel übergeht. Das RS232 MBI dagegen kann auch Daten empfangen. Da Sie das Centronics-Interface besitzen, gibt es keine Möglichkeit, Daten zu empfangen.

SONDERBARES IM SUCA EDITOR/ ASSEMBLER

Ich möchte Sie nur darauf hinweisen, daß das Spiel 'Freddy' nicht mit dem E/A der Firma SUCA läuft. Das Programm meldet sich nach kurzer Zeit, daß die Diskette eine Raubkopie sei. Dies tritt selbst dann auf, wenn dies nicht der Fall ist.
Mike Rohrmoser, Essen

Vielen Dank für den Hinweis. Wahrscheinlich klären sich nun einige Ungereimtheiten, die die User mit dem Spiel hatten.

CLUB-ECKE

Auch dieses Mal darf die Seite, auf der sich neugegründete oder schon bestehende TI-Userclubs vorstellen können, nicht fehlen. Die Einleitung bildet die in Stuttgart entstandene Gruppe 'TI-Club Errorfree'.

Nach langem Warten auf eine TI-Clubgründung im Großraum Stuttgart und Pforheim haben wir uns am 21.6.1986 entschlossen, einen eigenen TI-Club zu gründen.
Nach langem Überlegen einigten wir uns auf den Namen:
TI-Club Errorfree
Wir sind bisher zwar nur zwei Mitglieder, suchen jedoch weitere Mitglieder (auch Anfänger mit Extended Basic).
Aktivitäten sind geplant:
1. *Programmpool für Mitglieder*
2. *Informationsaustausch*
3. *Problemlösungen*
4. *Tel. oder schriftliche Kontakte (für später sind Clubtreffen geplant).*
5. *Später bei genügender Beteiligung eine Clubzeitung.*

Wir haben zwei komplette Anlagen zur Verfügung, wobei eine Anlage zusätzlich über DFÜ verfügt.
Info gegen 80 Pfennig Briefmarke oder telefonisch
Kontaktadressen:
Manfred Kraus
Vaihinger Str. 64
7132 Illingen/Württemberg
Tel.: 07042/24415
oder
Heinz Lücke
Mörikestraße 30
7147 Hochdorf/Enz
DFÜ-Tel.: 07042/77629

Den Anschluß daran bildet der Club '2099er', der seine Aktivitäten schon seit längerer Zeit ausübt. Diese, auch im Sommer aktive, Gruppe bietet die inzwischen schon dritte Ausgabe ihrer Clubzeitung an.

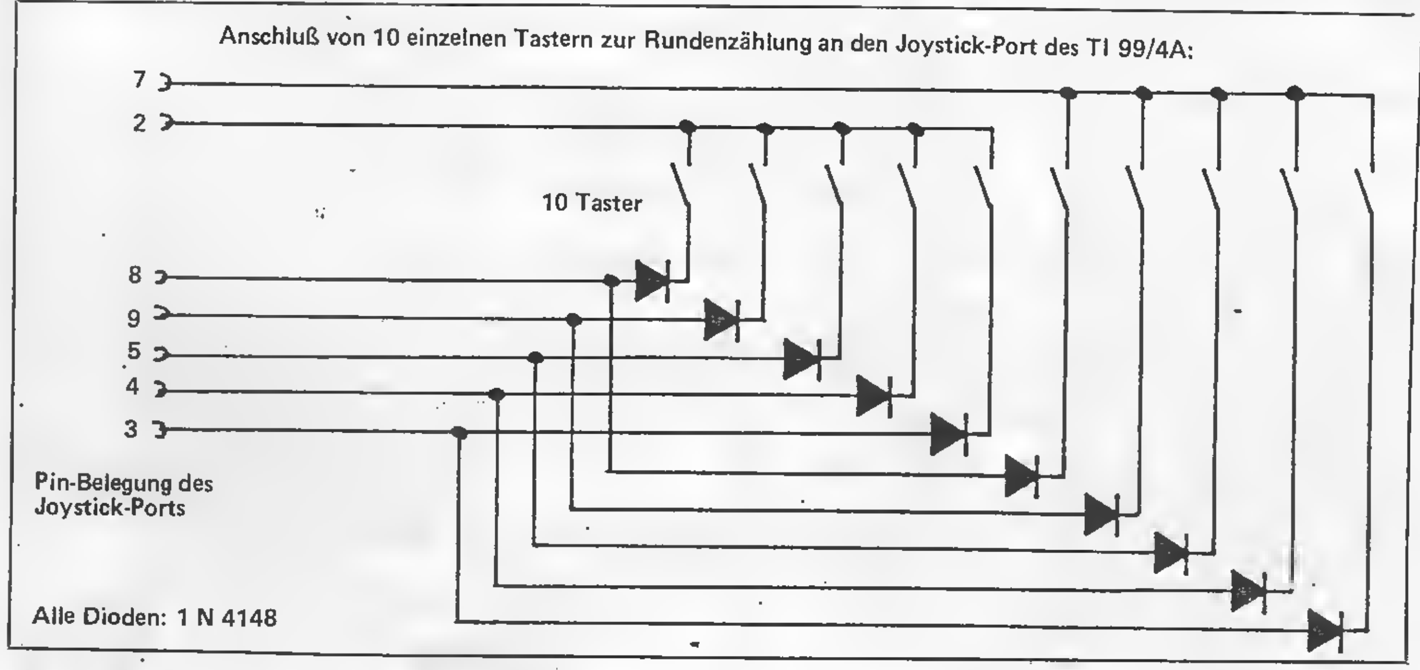
Nun ist es wieder soweit, die dritte Ausgabe der 2099er ist fertig! Diesmal gibt es auch noch ein paar Neuerungen. So kommt der Inhalt jetzt nicht nur, aber auch, vom Club der Toten sondern vom Rex Soft Club!
Die 2099er ist wieder voll mit Informationen, Berichten, Listings. Der "Preis" ist der alte geblieben, Ihr erhaltet die 2099er für zwei 50-Pfg.-Briefmarken für Porto und Verpackung.
In der letzten TI-REVUE unterstellte man uns Clubs, in der Sommerzeit wenig aktiv zu sein, aber das stimmt nicht, zumindest was den CDT und den Rex Soft Club betrifft.
Informationen über diese beiden Clubs und ihre Aktivitäten stehen in der 2099er Nr. 3, die Ihr Euch unbedingt besorgen müßt, lieber heut als morgen.
Unsere Adresse:
2099er
Am Hang 5
2054 Geesthacht

Das Finale der Clubseite wird von einem überaus aktiven Club, dem'TI 99'er Workshop Rheinland', gestaltet. Dieser plant ernsthaft die Ausführung eines nationalen Treffens aller TI-Clubs in Deutschland. Ob dieses jedoch zustande kommt, entscheidet unter anderem das Interesse aller bisher bestehenden Arbeitsgruppen. Ebenfalls stellt der 'TI 99'er Workshop Rheinland' das geplante Programm dieses nationalen Treffens vor.

Angeregt von unseren holländischen Freunden, die am 17.5. in Ede, Holland ihr nationales TI-Treffen veranstalteten, sowie vom Ergebnis einer Umfrage auf einer Demo-Show in Wuppertal, haben wir, der TI 99er Workshop Rheinland, uns entschlossen, bei entsprechendem Interesse der deutschen TI-Clubs und TI-User, ein solches nationales Treffen gegen Ende des Jahres auch in der Bundesrepublik Deutschland zu organisieren.
In der Umfrage in Wuppertal sprachen sich samt und sonders alle Befragten für eine solche Veranstaltung aus. Auch die Clubs, mit denen wir bis jetzt Kontakt aufgenommen haben, äußerten sich sehr positiv diesen Plänen gegenüber.
Es ist geplant, diese Veranstaltung (der zentralen Lage wegen) in Köln, Frankfurt oder Düsseldorf stattfinden zu lassen, wobei Köln aus naheliegenden Gründen favorisiert ist. Die Stadtverwaltung der Stadt Köln hat sich auch schon positiv zu diesem Vorhaben geäußert. Da die Ausrichtung eines solchen Treffens doch eine ganze Menge Geld erfordert, haben wir schon Kontakt mit einigen Firmen aufgenommen, um von dort eventuell einen Teil des Werbeetats locker zu machen.
Hier die Adresse an die Ihr Euch wenden könnt:
TI 99er Workshop Rheinland
Dept. Allgemein & Software
c/o Mike Heuser
Karl-Marx-Allee 18
5000 Köln 71
Tel. 0221/703979 v. 19–22 Uhr
Hier noch ein kleiner Tip am Rande. Unter der Nummer 0228/658232 ist ab sofort die Mailbox des TI 99er Workshop Rheinland, vorerst im Test(Hand)betrieb online. Ihr könnt dort von 17–22 Uhr anrufen. Sysop Achim freut sich auf Euren Anruf.
Seid aber bitte nicht sauer, wenn es doch mal einen Fehler im Programm geben sollte.

DIREKTE JOYSTICK-ABFRAGE AUS ASSEMBLER

In der letzten Zeit wurde ich einige Male darauf angesprochen, wie man beim TI 99/4A den Joystick-Port direkt aus Assembler abfragen könne, ohne Umweg über die Hilfsroutine KSCAN. Bevor ich jedoch dazu komme, noch eine Anmerkung zu meinem Artikel „Wie findet der TI seine Programme" in der TI-REVUE 7/86.



Anschluß von 10 einzelnen Tastern zur Rundenzählung an den Joystick-Port des TI 99/4A:

Ein aufmerksamer Leser hat mich darauf aufmerksam gemacht, daß einmal auch im ROM des Modulportes ein Pointer zu einer Programmliste stehen kann und weiter das Byte mit der Versionsnummer >6001 durchaus vom bisherigen Betriebssystem beachtet wird. Ist hier nämlich bei einem GROM-Modul das höchstwertige Bit gesetzt, so führt der TI 99/4A noch vor Erscheinen des Titelbildes das Programm in diesem Modul (über GROM-Adresse >6010) aus. Dies ist natürlich richtig, aber in meinem Bericht habe ich nur vom DSRLNK des Betriebssystems geredet, und dieses hat mit dieser Funktion nichts zu tun, sondern diese ist extra in die Powerup Routine des GROM 0 einprogrammiert. Ebenso übrigens wie die Möglichkeit der reinen ROM-Module. Letztere ist wohl von einem cleveren Programmierer eingesetzt worden, denn dies ermöglichte Fremdherstellern den einfachen Bau von Modulen. So hat Texas Instruments dies dann auch bei späteren Modellen des TIO 99/4A (Bildschirmmeldung Copyright 1983) wieder entfernt. Auf diesen laufen die reinen ROM-Module nicht mehr.

Nun aber zurück zur Joystick-Abfrage. Um hier in Assembler entsprechend vorgehen zu können, müssen wir uns erst einmal etwas mit den Innereien des TI 99/4A beschäftigen; insbesondere natürlich mit der Tastaturabfrage.

Diese Tastatursteuerung geschieht im wesentlichen über einen TMS 9901, einen programmierbaren Baustein, der über CRU-Adressen angesprochen wird. Dieser IC besitzt nun diverse Eingänge und einige Ausgänge. Drei dieser Ausgänge werden benutzt, um über einen zusätzlichen Adressdekoder insgesamt 6 Steuerleitungen für die Tastatur und zwei Steuerleitungen für die Joysticks zu betätigen. Eine zusätzliche Steuerleitung ist für die Alpha Lock Taste vorhanden, die soll uns aber erst später interessieren.

Diese Steuerleitungen sind nun jeweils mit mehreren Tasten verbunden (im Joystick mit jeweils allen Kontakten). Um nun die entsprechenden Tasten bzw. Joystick-Kontakte herausfiltern zu können, sind dafür 9 Eingangsleitungen vorhanden, jeweils ebenfalls mit mehreren Tasten, aber natürlich anderen Tasten als die Steuerleitungen, verbunden. Für die Joysticks werden davon nur 5 benötigt. Aber zu den genauen Angaben kommen wir später noch. Zuerst müssen wir uns einmal damit beschäftigen, wie wir die einzelnen Steuerleitungen einschalten.

Dieses Einschalten erfolgt über ganz normale CRU-Befehle. Wie bekannt, muß dazu in R12 die CRU-Basisadresse vorhanden sein. Für unsere Steuerleitungsregister ist dies die Adresse >0024. Wie wir vorhin gesehen haben, müssen wir jeweils insgesamt 3 Bits bestimmen. Es stehen uns damit die folgenden 8 verschiedenen Werte zur Verfügung, welche jeweils mit folgenden Tasten verbunden sind:

0: Enter Shift Leerstelle = Function Control
1: 0 9 S L > 2 W X
2: I 8 D K < 3 E C
3: U 7 F J M 4 R V
4: Y 6 G H N 5 T B
5: P 0 A ; / 1 Q Z
6: Joystick A
7: Joystick B

Ich hoffe, das stimmt alles, denn außer den Joysticks habe ich die Tastaturabfragen noch nicht selber

ausprobiert, sondern die Werte hier wurden aus Schaltbildern ermittelt. Deshalb habe ich die Überschrift auch nur bezogen auf die Joysticks gewählt.
Vorerst bleiben wir also bei den Joysticks. Um diese abzufragen, müssen wir also erst einmal die richtigen Steuerleitung aktivieren.
Dies erfolgt für Joystick A:

```
LI      R12,>0024
LI      R4,>0600
LDCR    R4,3
```

Achtung: Da nur 3 CRU-Bits geschrieben werden, ist LDCR ein Byte-Befehl und so muß der Kennwert in R4 in das höherwertige Byte geschrieben werden.
Für Joystick B gilt dementsprechend

```
LI      R12,>0024
LI      R4,>0700
LDCR    R4,3
```

Nun müssen wir uns noch mit dem Abfragen der einzelnen Kontakte beschäftigen. Auch hier sind ja wieder verschiedene Tasten der Tastatur und die Joystick-Kontakte zusammengeschaltet. Auf den folgenden CRU-Adressen sind jeweils die angegebenen Tasten verbunden:

```
>0006:  / N = > < M
        Feuerknopf
>0008:  ; H Leerstelle L K J
        Links
>000A:  P Y Enter  O I  U
        Rechts
>000C:  0 6 9 8 7 Abwärts
>000E:  1 5 Funktion 2 3 4
        Aufwärts und Alpha
        Lock
>0010:  A G Shift  S D F
>0012:  Q T Control  W E R
>0014:  Z B X C V
```

Wir benötigen für die Joysticks also nur die CRU-Adressen> 0006 bis >000E und jetzt müssen wir wieder etwas aufpassen, denn bekanntlich muß beim TI 99/4A ja die CRU-Basisadresse in R12 anders errechnet werden als für die 1-Bit-Befehle TB SBO und SBZ. Wir können daher die Abfrage auf verschiedene Weise realisieren. Zuvor aber müssen wir noch kurz den Wert bestimmen, den eine gedrückte Taste annimmt, wenn die passende Steuerleitung eingeschaltet ist.
Die Steuerleitungen haben normalerweise den Wert H, also ca. 4 bis 5 V. Wenn eine selektiert wird, wie wir es oben für die Joystickleitungen gemacht haben, geht sie auf den Wert L, d.h., auf ca. 0 Volt. Die Taste verbindet den Eingang des TMS 9901, der uns zum Abfragen dient, dann mit der Steuerleitung, d.h., bei der entsprechenden CRU-Abfrage ist das Bit nicht gesetzt, wenn die Taste gedrückt ist. Ist die Taste nicht gedrückt, ist das Bit 1.

Fragen wir nun also unseren Joystick ab. Dabei soll erst einmal jede Taste einzeln abgefragt werden:

```
CLR     R12     *Basisadresse 0
TB      3       *Feuerknopf
JNE     FEUER*Sprung gedrückt
TB      4       *Links
JNE     LINKS
TB      5       *Rechts
JNE     RECHTS
TB      6       *Runter
JNE     RUNTER
TB      7       *Hoch
JNE     HOCH
```

Die Label für die Sprünge sind dabei je nach Programm entsprechend auszuführen.
Nun können wir aber auch, um nicht immer die CRU-Adresse in R12 ändern zu müssen, das ganze mit der Basisadresse für die Steuerleitungs-Selektierung durchführen. Dies sähe dann so aus:

```
LI      R12,>0024
TB      -15     *Feuerknopf
JNE     FEUER*Sprung gedrückt
TB      -14     *Links
JNE     LINKS
TB      -13     *Rechts
JNE     RECHTS
TB      -12     *Runter
JNE     RUNTER
TB      -11     *Hoch
JNE     HOCH
```

Nun läßt sich das alles natürlich noch eleganter lösen, denn warum sollen wir jegliches Bit einzeln testen. Die folgenden Zeilen zeigen, wie sich mit einem Schritt alle Tasten einlesen lassen:

```
LI      R12,>0006 *Basisadresse
CLR     R4
STCR    R4,5
```

Nun sind im höherwertigen Byte von R4 die Bits entsprechend den gedrückten Tasten gesetzt, nach folgendem Muster:

```
MSB           LSB
8 7 6 5 4 3 2 1
0 0 0 U D R L F
```

U=Hoch, D=Abwärts, R=Rechts, L=Links und F=Feuerknopf

Das Register wird dann im Programm wie gewünscht weiterverarbeitet. Hier bieten sich dann ganz besonders die Schiebebefehle an, um entsprechende Sprünge auszuführen, oder die Maskenvergleiche COC bzw. CZC. Damit lassen sich dann auch Abfragen nach zwei zusammen gedrückten Tasten verwirklichen.
Bleibt uns zum Schluß nun doch noch eine gezielte Anfrage der Tastatur. Bekanntlich geht ein Joystick nicht "hoch", wenn die Alpha-Lock-Taste gedrückt ist. Dies ist übrigens durch die Schaltung des TI 99/4A bedingt und kann softwaremäßig nicht geändert werden. Es bietet sich hier aber an, diese Taste getrennt abzufragen und bei gedrückter Alpha-Lock-Taste eine entsprechende Meldung auf den Bildschirm zu zaubern.
Die Steuerleitung der Alpha-Lock-Taste ist das CRU-Bit >002A. Also können wir diese Taste wie folgt abfragen:

```
CLR     R12
SBZ     >0015 *Steuerleit. ein
NOP           *Zeitverlust
TB      7
JEQ     NAL*Sprung wenn nicht
*Hier Anzeige einbauen
NAL     SBO >0015
*Weiter im Programm
```

Achtung: Auf gar keinen Fall am Ende das SBO >0015 vergessen, sonst kommt es bei weiteren Abfragen bei gedrückter Alpha-Lock-Taste zu ziemlichem Unfug.
Das war es dann für heute erst einmal wieder. Auf unserer kleinen Zeichnung ist zu sehen, wie man anstelle des Joysticks auch einzelne Taster zur Dateneingabe verwenden kann. Diese Taster werden an den Joystickport angeschlossen.
Ich selber benutze so etwas für ein Rundenzähl- und Zeitmeßprogramm für ferngesteuerte Modellrennwagen. Hier hat sich der TI seit Jahren als zuverlässige Hilfe bewährt.

Heiner Martin

UNSER ☏ TELEFONSERVICE ☏

Leserbriefe und Fragen sind uns stets willkommen.
Wir beantworten sie entweder direkt oder auf der Leserbriefseite



ACHTUNG !!!
Wie immer steht unseren Lesern unser Telefon-Service zur Verfügung! Jeden Dienstag von 15 bis 19 Uhr. Für technische Fragen: 0731/33 220 und für Listings/Programme: 089/129 80 13

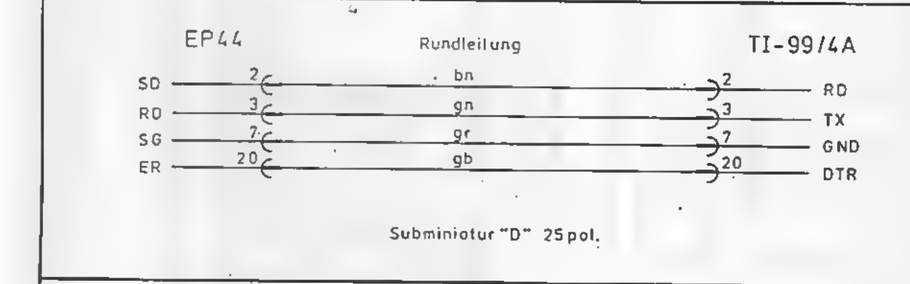
NACHTRAG ZUM MULTI FUNKTIONS KATALOG

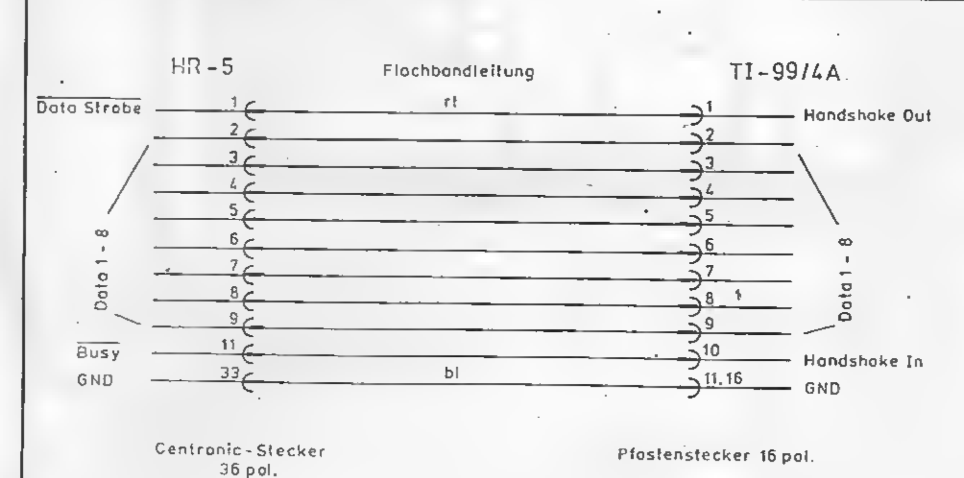
Bei diesem Programm hat uns die Herstellung ein Schnippchen geschlagen: Es wurde ein ganzer Abschnitt aus dem Manuskript weggelassen. Dieser ist aber zum Verständnis enorm wichtig und so hier nochmal der entsprechende Text:

MIC-GEHEIMNISSE

MIC ist ein Extended-Basic-Programm mit darin „versteckten" Maschinensprache-Unterprogrammen (Lesen und Schreiben im VDP-RAM und Lesen und Schreiben von Diskettensektoren). Die Unterprogramme werden also weder durch CALL-LOAD-Aufrufe noch durch Poken aus DATA-Zeilen geladen. Sie sind auf ihren Plätzen, sobald das Basicprogramm geladen ist. Damit die Unterprogramme in einem von der CPU direkt erreichbaren Speicherbereich liegen, muß eine Speichererweiterung angeschlossen sein.
Derartige Programme findet man in letzter Zeit häufiger. Deshalb für Interessierte hier ein paar Hinweise darauf, wie man so etwas macht. Im Ur-Basicprogramm werden einige Zeilen vorgesehen, die nur als Platzhalter dienen, am einfachsten Kommentarzeilen. Der so reservierte Platz im Speicher muß ausreichend sein, die Unterprogramme aufzunehmen. In MIC waren die Zeilen 180, 190 und 200 ursprünglich lange Kommentarzeilen. Das vorbereitete Basicprogramm wird geladen, und mit Hilfe von DEBUG wird nachgesehen, ab welcher Adresse diese Platzhalter im Speicher stehen. Dann kann mittels einer entsprechenden AORG-Anweisung im Assemblerprogramm dafür gesorgt werden, daß den Unterprogrammen die richtigen Plätze zugewiesen werden. Mit DEBUG wird weiterhin geschaut, auf welchen Speicherplätzen in der Zeilenliste die Adressen der Platzhalter-Zeilen stehen und welche Adresse eine Ersatzzeile hat, die formal statt der überschriebenen Platzhalter-Zeilen dienen kann. Als ein solcher Ersatz dient in MIC die kurze Kommentarzeile 160. Die Adressen nach den Platzhalter-Zeilennummern (in MIC 180, 190 und 200) werden später mit der Adresse der Ersatzzeile (in MIC 160) überschrieben.
Es muß dafür gesorgt werden, daß die Unterprogramme bei CALL-LINK-Aufrufen auch gefunden werden. Deshalb wird vom Basicprogramm aus die REF/DEF-Tafel am Ende des LOW MEMORY durch Poken erstellt. Die Anfangsadresse der REF/DEF-Tafel muß nach Adresse 8196 geschrieben werden. In MIC geschieht das alles in Zeile 280 unter Verwendung der DATA-Zeilen 220 und 230.

Beim Entstehungsprozeß sollte nach Änderungen im Ur-Basicprogramm dieses wieder in eine im Speicher wohlgeordnete Form gebracht werden, z.B. mit der Anweisungsfolge
SAVE DSK1.UR_MIC





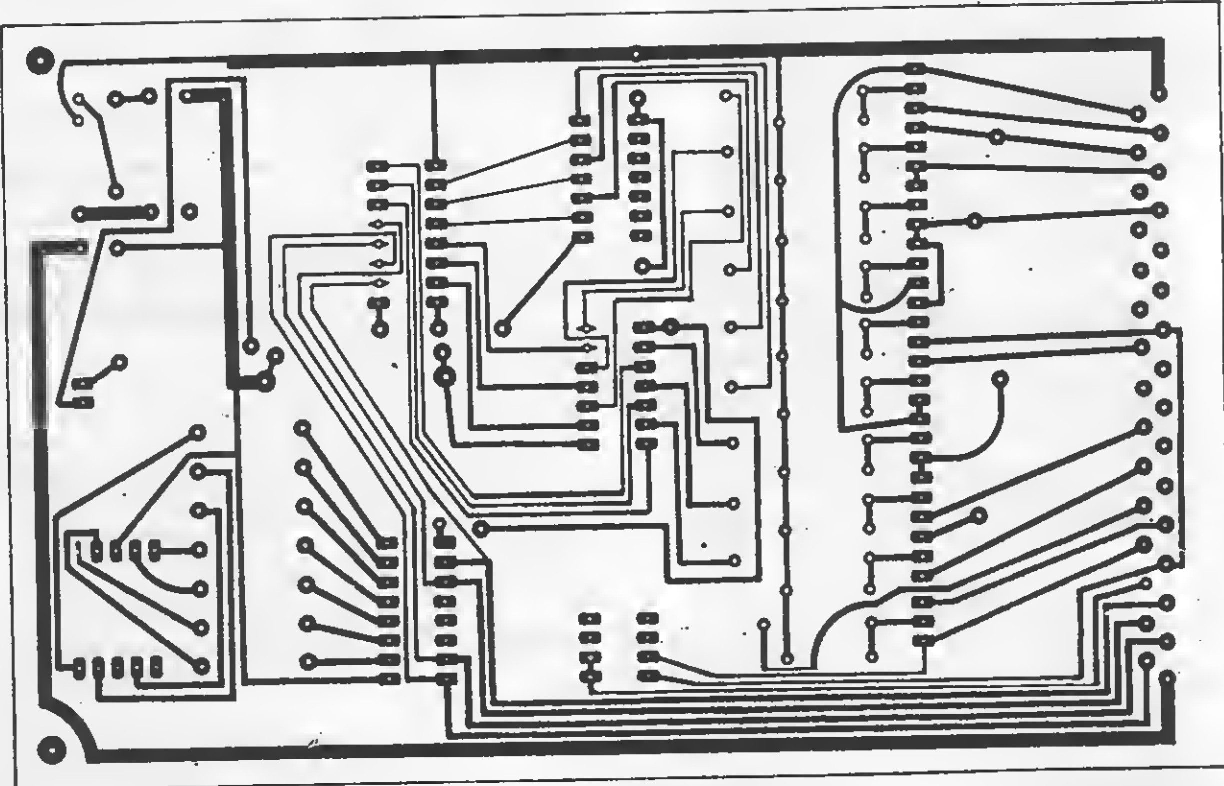
DRUCKER-ANSCHLUSS

Immer wieder erreichen uns Anfragen nach der Ausführung von Anschlußkabeln für die verschiedenen Drucker. Die hier vorgestellten Verbindungen sind zwar für zwei bestimmte Druckertypen, können aber durchaus allgemein verwendet werden. Der Anschluß der EP 44 zeigt die Verbindung mit der RS232-Schnittstelle des TI-99/4A und das Kabel für den HR-5 ist für die PIO gedacht.

BC-DEZIMAL-WANDLER

Der Einstieg in die Hardware wird dem Besitzer des TI 99/4A nicht besonders leicht gemacht. Will man z.B. Werte oder auch nur Schalterstellungen ins Programm einlesen, so stößt man bei den vorhandenen Ports schnell an einige Grenzen. Der einzige herausgeführte und direkt dafür zu verwendende Eingang ist der Joystick-Port. Doch pro Programmzeile läßt sich nur ein Schalter abfragen (Joyst. 1 oder Joyst. 2). Im Geschäft zu erstehende Erweiterungen, die solche Möglichkeiten bieten, sind recht teuer.

Doch auch die „nackte" Konsole verfügt noch über einen Port, der für solche Zwecke geeignet ist, den Key-Board-Port. Er ist allerdings nicht aus dem Gehäuse herausgeführt. Ihn wollen wir bei der hier vorgestellten Schaltung verwenden. Um an ihn heranzukommen, müssen wir wohl oder übel das Gehäuse öffnen. Ist dies geschehen, so braucht man nur noch die Verbindungsstelle zwischen Hauptplatine und Tastaturplatine zu suchen.
Die Verbindung erfolgt über eine 15adrige Flachbandleitung, die an der Tastaturplatine angelötet und mit der Hauptplatine über einen 15poligen Stecker verbunden ist. Dieser Stecker läßt sich ohne weiteres abziehen, allerdings sind dafür und für spätere Arbeiten alle Baugruppen (Hauptplatine, Tastaturplatine und Netzteil) aus dem Gehäuse auszubauen.
Der Key-Board-Port besitzt 15 Pins, deren Numerierung auf dem Stecker abzulesen ist. Benötigt werden davon 12 Anschlüsse, doch ist es zweckmäßig alle 15 und zusätzlich die Masse, die man sich vom Blechgehäuse der Hauptplatine oder direkt vom Netzteil holt, herauszuführen. Dadurch sind dann auch für spätere Anwendungen alle nötigen Anschlüsse von außen zu erreichen.
Benötigt wird also ein Stecker mit mindestens 16 Anschlüssen (z.B. Steckverbinder SUB D 25polig), der ins Gehäuse eingebaut und mit dem Key-Board-Port verbunden wird.



Im Gehäuse ist für Einbau und Verdrahtung genügend Platz vorhanden. Als Einbauposition für den Stecker habe ich die linke Seitenwand schräg über dem Joystickanschluß gewählt (ca. 15 mm von oben und ca. 55 mm von hinten).
Bei der Ausführung der Verdrahtung sind der Phantasie keine Grenzen gesetzt. Man kann z.B. die Flachbandleitung von der Tastaturplatine ablöten und die Anschlüsse auf eine kleine Verteilerplatine legen, von der dann die Leitung für Tastatur und Stecker abgehen. Wichtig ist dabei nur die Einbehaltung der Reihenfolge und Bezeichnung der Anschlüsse, d.h., Anschluß 1 am Key-Board-Port muß auch auf dem Stecker Anschluß 1 sein usw. Die Masse wird auf einen der freien Anschlüsse des Steckers gelegt. An diesen Port läßt sich z.B. auch ein separater Tastenblock für die Eingabe von Zahlen anschließen.

FUNKTIONSBESCHREIBUNG

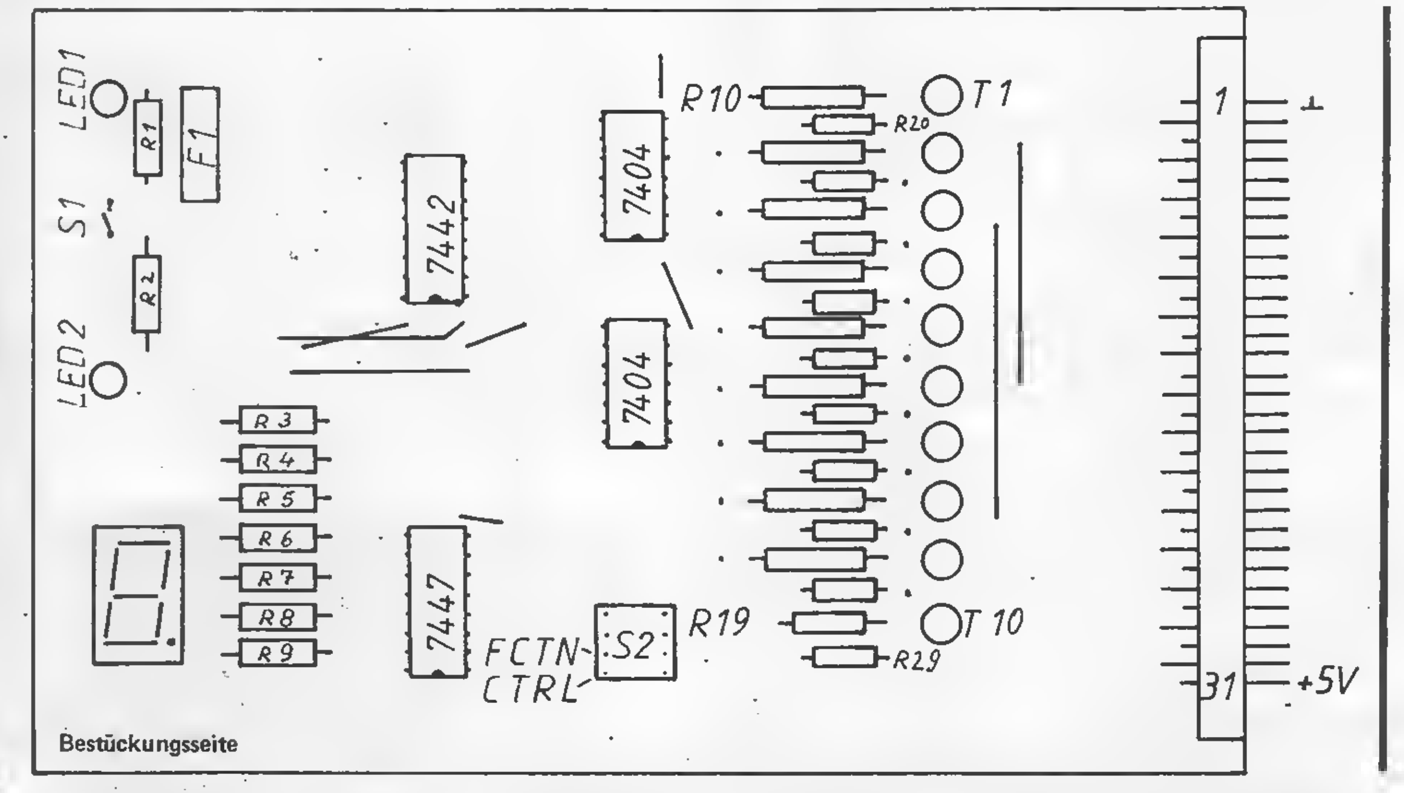
Das Grundprinzip der Wandlerschaltung ist recht einfach. Ihr Eingang besteht aus den Anschlüssen A, B, C und D. Es können nun z.B. 4 Schalter angeschlossen (+5 V=geschlossen; 0 V=geöffnet) oder Zahlenwerte von 1 – 9 im BCD-Code ($A = 2^0=1$; $B = 2^1=2$; $C = 2^2=4$; $D = 2^3=8$) angelegt werden. Die angelegten Potentiale gelangen an den IC 7442, der ein BCD-Dezimal Decoder ist. Dieser Baustein hat 10

Ausgänge. Liegt an den Eingängen der Wert 0 an, so liegt der 0-Ausgang auf Masse. Die grüne an ihm angeschlossene Leuchtdiode zeigt durch ihr Licht an: Keine Eingabe bzw. alle Schalter offen.
Liegt am Eingang z.B. der Wert 4 an (C=+5 V bzw. Schalter C geschlossen), so liegt der Ausgang 4 auf Masse. Dieses Potential wird durch einen Inverter (IC 7404) in ein High=+5 V umgewandelt und gelangt an den Spannungsverteiler R 13/R 23.
Der Transistor T 4 schaltet durch und verbindet die Anschlüsse 13 und 2 des Key-Board-Ports, die sonst durch die Taste D der Tastatur verbunden würden. Bei der Abfrage der Tastatur über CALL KEY wird der dem Buchstaben D zugeordnete Wert ins Programm übergeben.
Es gibt nun 2 Möglichkeiten, um im Programm wieder den angelegten Zahlenwert (den betätigten Schalter), also in diesem Beispiel die 4 (Schalter C) zu erhalten:

1. per Hardware
Der Schalter S 2-CTRL auf der Platine wird nach dem Starten des Programms geschlossen. Dieser Schalter verbindet die Anschlüsse 4 und 6 des Key-Board-Ports und simuliert die Betätigung der Taste CTRL.
Mit der Programmzeile CALL KEY (4, K, S) wird für K, in diesem Beispiel, der Wert 4 gespeichert.
Ebenso wird jeder andere Wert im Bereich von 1 – 9 gleich richtig ins Programm übernommen. Bevor Eingaben über die Tastatur erfolgen, ist der Schalter S 2-CTRL zu öffnen, um die Funktion der Tastatur nicht zu behindern.
2. per Software
Will man die hier vorgestellte Schaltung nur in der noch später erläuterten Grundversion bauen, so fragt man den Port mit dem Befehl CALL KEY (O, K, S) ab und zieht von dem nun im ASCII-Code übernommenen Wert in einer weiteren Programmzeile den Wert 64 ab. Man erhält so ebenfalls den ursprünglich angelegten Wert.

LUXUS- UND GRUNDVERSION DER SCHALTUNG

Die Grundversion ist im oberen Teil des Schaltplanes dargestellt. Sie erfüllt schon voll die beschriebene Funktion. Jedoch müssen die übernommenen Werte noch, wie oben beschrieben, im Programm „verarbeitet“ werden oder man baut separat einen Schalter ein, der die Pins 4 und 6 des Key-Board-Ports verbindet. Die für den Aufbau der Grundversion benötigten Bauteile lassen sich aus dem oberen Teil des Schaltplans ersehen. Es werden dann nur diese Teile der Stückliste benötigt.
Die Luxusversion der Schaltung enthält neben der Grundversion noch einige recht vorteilhafte Zusätze.
Mit dem Schalter S 2 lassen sich die Kontakte der Tasten CTRL und FCTN betätigen. Das Schließen von S 2-CTRL bewirkt, wie schon beschrieben, die rein hardware-mäßige Umwandlung der Werte.
Betätigt man S 2-FCTN, so wird beim Anlegen von +5 V an den Eingang “CLEAR“ das Programm unterbrochen. Dieser Eingang der Schaltung läßt sich z.B. zum Anschluß einer “Run-Stop-Taste“ verwenden oder ermöglicht es, den Computer im Programmablauf durch selbst entwickelte Peripherie zu stoppen.
Läßt man den Schalter S 2-FCTN geöffnet, so läßt sich beim Anlegen von +5 V an den Eingang “CLEAR“ der Wert, welcher der Taste 4 zugeordnet wird, ins Programm einlesen. Er kann dann spearat abgefragt und z.B. als Fehlermeldung der Peripherie herangezogen werden.
Der untere Teil des Schaltplanes zeigt einen weiteren Zusatz der Luxusversion. Er stellt eine dezimale Anzeigeeinheit der dual am Eingang anliegenden Werte, dar. Auf dieser Anzeige lassen sich Werte jederzeit ablesen. Man braucht sie also nicht extra durch “PRINT“ auf dem Bildschirm abzufragen.

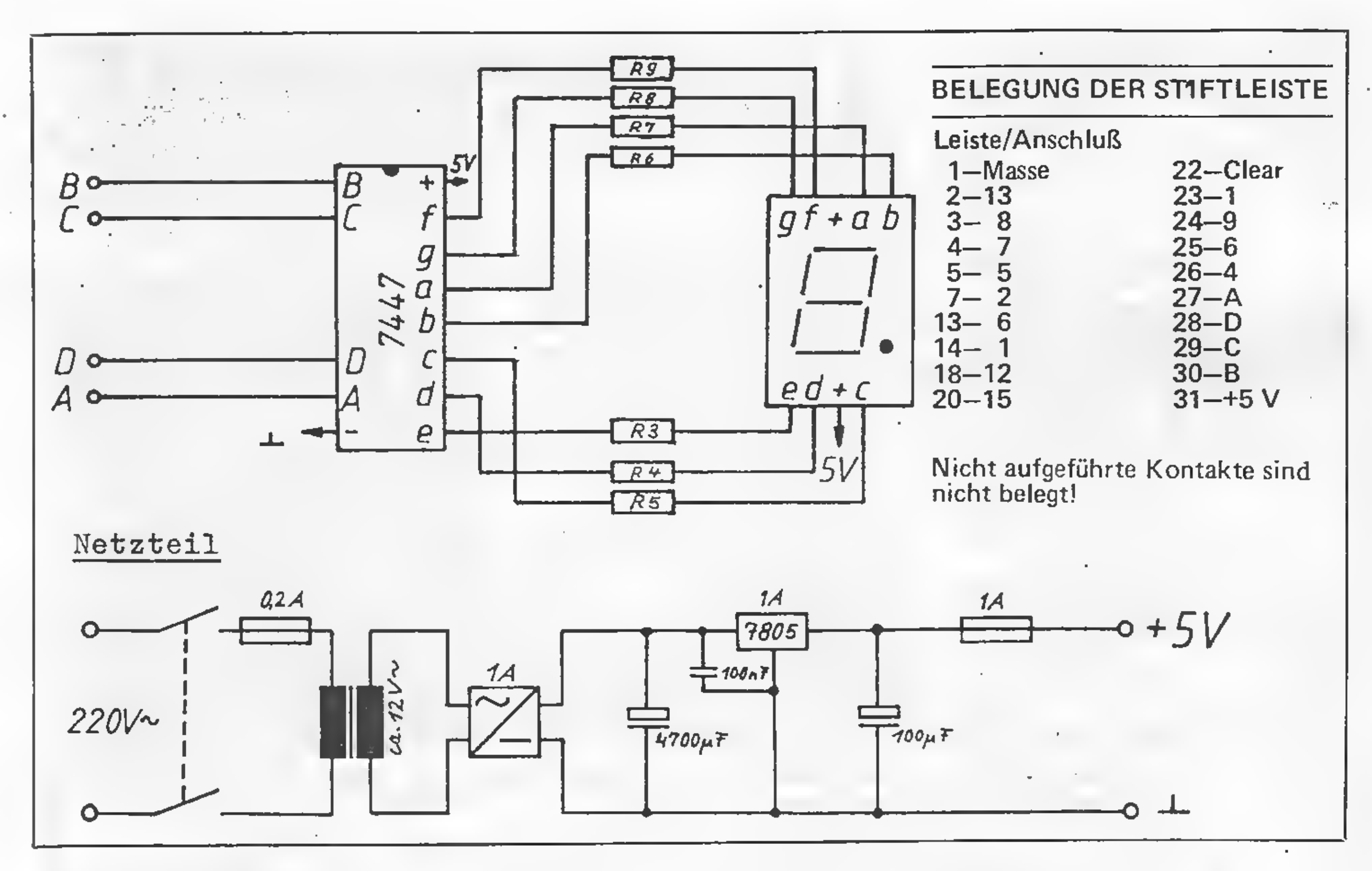


Bestückungsseite

AUFBAU

Entscheidet man sich für die Luxusversion, so kann man beim Aufbau der Schaltung auf das abgedruckte

SERVICE



BELEGUNG DER STIFTLEISTE

Leiste/Anschluß	
1–Masse	22–Clear
2–13	23–1
3– 8	24–9
4– 7	25–6
5– 5	26–4
7– 2	27–A
13– 6	28–D
14– 1	29–C
18–12	30–B
20–15	31–+5 V

Nicht aufgeführte Kontakte sind nicht belegt!

Schaltpläne und Tabellen

TABELLE KEY-BOARD-PORT

```
Taste: /  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
------------------------------------------------------------
   A                           o--------------o         T1
   B                  o-----o                           T2
   C         o--------o                                 T3
   D         o--------------------------------o         T4
   E         o-----------o                              T5
   F      o-----------------------------------o         T6
   G                        o-----------------o         T7
   H                        o--------------o            T8
   I         o-----------------------------------------oT9
------------------------------------------------------------
CTRL               o--------o
FCTN               o---------------o
   4      o------------------------o                    T10
```

Erläuterung zur Tabelle:

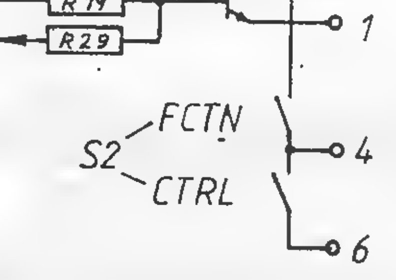
Die Zahlen 1–15 stellen die PIN's des Key-Board-Ports dar. Mit den Buchstaben A–I sind die entsprechenden Tasten der Tastatur gemeint. Bei ihrer Betätigung werden die durch die Strichlinien dargestellten Verbindungen geschaffen.
In der Schaltung verbindet der in der gleichen Reihe stehende Transistor die entsprechenden PIN's.

Platinenlayout zurückgreifen.
Es handelt sich bei der Platine um eine sogenannte Europakarte (100x160 mm), die als Einschub ausgelegt ist. Der Anschluß erfolgt über eine 31polige Stiftleiste. Will man nicht die Einschubversion bauen, so können die Verbindungsadern natürlich auch direkt eingelötet werden.
Neben den im Schaltplan dargestellten Bauteilen sind auf der Platine noch der Schalter S 1 für Ein/Aus, eine Feinsicherung F 1 und eine Leuchtdiode LED 1 mit Vorwiderstand zur Betriebsanzeige vorhanden. Für die IC's sollte man Sockel verwenden, um ihre vorzeitige Zerstörung beim Einlöten zu vermeiden. Die angegebenen Brücken sind mit geeignetem Draht (z.B. Silberdraht) herzustellen.
Soll die Schaltung nur in der Grundversion zur Anwendung kommen, so kann man sich nach dem oberen Teil des Schaltplans selbst ein Platinenlayout erstellen oder man baut das Ganze auf einer vorgefertigten Platine mit Lötstreifen auf.

ANSCHLUSS- UND FUNKTIONSÜBERPRÜFUNG

Sind alle Bauteile eingelötet, so überprüft man noch einmal anhand des Schaltplanes ihre korrekte Einbaulage und die Richtigkeit der Anschlüsse.

SERVICE



STÜCKLISTE

Transistoren:
10 x BC 546 / T 1–T 10
IC's:
2 x 7404 (je 6 Inverter)
1 x 7447 (BCD-7 Segment Decoder)
1 x 7442 (BCD-Dezimal Decoder)
LED's:
1 x rot; 5 mm; 1,6 V / LED 1
2 x grün; 5 mm 1,6 V / LED 2
LED-Anzeige:
1 x D 350 PA (7-Segmentanzeige; 13 mm Ziffernhöhe; 1,6 V; gem. Anode)
Widerstände:
9 x 270 Ohm / R 1 – R 9
10 x 30 kOhm / R 10 – R 19
10 x 33 kOhm / R 20 – R 29
Sonstiges:
1 x Sicherungshalter mit Feinsicherung (0,5 A) / F 1
1 x 4pol. DIL Schalter / S 2
1 x Stiftleiste; 31pol.; 2,5 mm Raster
1 x Aus-Schalter / S 1

Stückliste Netzteil:
Halbleiter:
1 x Spannungsregler 7805; 1 A
1 x Brückengleichrichter B 40 C 1500/1000
Kondensatoren:
1 x 4700 uF / 25 V (Elko)
1 x 100 uF / 10 V (Elko)
1 x 100 nF
Sonstiges:
1 x Trafo; prim. 220 V; sek. ca. 12 V
2 x Sicherungshalter mit Feinsicherung (0,2; 1 A)
1 x Ausschalter 2pol.; 220 V

Anmerkung:
Das Netzteil ist für spätere Erweiterungen absichtlich stärker ausgelegt worden.

Nun legt man die Betriebsspannung von 5 V an, LED 1 sollte leuchten, LED 2 und die 7-Segmentanzeige müssen noch dunkel sein. Legt man nun die Anschlüsse A – D auf Masse, so sollte LED 2 leuchten und die Anzeige eine 0 darstellen. Ist dies gegeben, stellt man die angegebenen Verbindungen zum Key-Board-Port und zur Masse her. In den Computer wird folgendes Programm eingegeben:

```
100 CALL KEY (4, K, S)
110 IF S = 0 THEN 100
120 PRINT K
130 GOTO 100
```

Man startet das Programm und schließt S 2-CTRL. Anschließend werden die Werte 1 – 9 im BCD-Code an die Anschlüsse A – D gelegt, wobei die angelegte Zahl auf der 7-Segmentanzeige und auf dem

Bildschirm sichtbar werden müßte. Sollten einige Werte zwar auf der Anzeige, nicht aber auf dem Bildschirm erscheinen, so sind die entsprechenden Transistoren zu drehen (Kollektor und Emitter vertauschen).
Nach Betätigen von S 2-FCTN und Anlegen von +5 V an den Anschluß CLEAR (Pin 22 Stiftleiste) sollte der Programmablauf gestoppt werden. Sind diese Funktionen alle gegeben, so funktioniert die Schaltung einwandfrei.

SPANNUNGSVERSORGUNG

Die Schaltung benötigt 5 V und ca. 300 mA. Es ist am besten, sich für die Versorgung ein separates Netzteil aufzubauen, wenn dies noch nicht vorhanden sein sollte. Der Aufbau ergibt sich aus Schaltplan und Stückliste und dürfte keine Schwierigkeiten bereiten. Unterbringen sollte man das Netzteil in einem Metallgehäuse, um das Magnetfeld des Trafos abzuschirmen. Die Auslegung des Netzteils läßt spätere Erweiterungen, sowohl vom Spannungswert hinter dem Trafo, als auch vom Ausgangsstrom, durchaus zu.

BCD-CODE

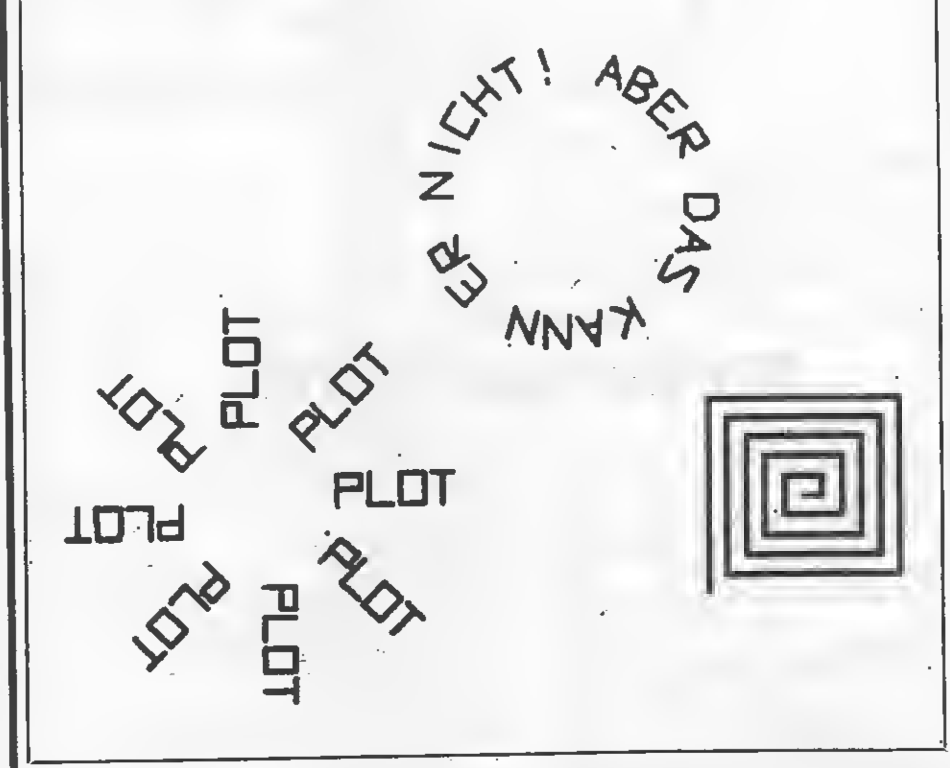
Dez.Zahl	D=8	C=4	B=2	A=1
0	–	–	–	–
1	–	–	–	+
2	–	–	+	–
3	–	–	+	+
4	–	+	–	–
5	–	+	–	+
6	–	+	+	–
7	–	+	+	+
8	+	–	–	–
9	+	–	–	+

ANWENDUNGEN

Bei der Anwendung der Schaltung sind der Phantasie keine Grenzen gesetzt. Man kann z.B. die Raumtemperatur über entsprechende Wandler erfassen und in den Computer zur Weiterverarbeitung einlesen.
Der beschränkte Bereich von 1 – 9 ist durchaus ausreichend dafür, da die Temperatur ja nur in bestimmten Grenzen schwankt. Der erfaßbare Bereich wäre z.B. von 16° – 24° C.
Weitere Anwendungen wären z.B. eine Alarmanlage; das Einlesen von in bestimmten Grenzen schwankenden Spannungs- oder Stromwerten oder die Positionsabfrage eines selbstgebauten Roboterarmes.

Peter Thorhauer

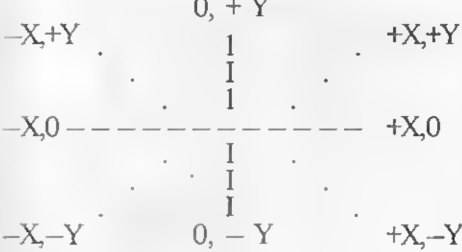
PLOTTER-SELBST GEBAUT (III)



Um den Plotter von BASIC aus auch nun betreiben zu können, stehen folgende Programme zur Verfügung, die mit CALL LINK ("Prgm",P1, P2,..) aufgerufen werden.
CALL LINK<("PLTR",X,Y,IP)
relativ Plotten
CALL LINK<("PLTA",X,Y,IP)
absolut Plotten
CALL LINK<("RESET")
Ursprung und Startwerte setzen
CALL LINK<("FAKTOR",XFAK, YFAK)
Skalierung definieren
Diese Programme nehmen dem Anwender die Arbeit ab, sich ausrechnen zu müssen, wie viele Plotschritte nun nötig sind, um den Stift um die gewünschte Strecke zu bewegen, außerdem erzeugen sie den Kode für die Richtung und den Zeichenstift. Die Plotprogramme

SERVICE

können durch Drücken einer Taste jederzeit abgebrochen werden, es erfolgt dann der Rücksprung ins BASIC.
Da in alle Richtungen unabhängig voneinander gezeichnet werden kann, stehen acht Richtungen zur Verfügung, in die der Stift ohne Interpolation gefahren werden kann: horizontal, vertikal und 45 Grad diagonal.



Die beiden Programme "PLTR" und "PLTA" arbeiten nach diesem Prinzip

CALL LINK<"PLTR",X,Y,IP)
Dieses Programm zeichnet relativ, d.h. der Stift wird von der augenblicklichen Position um den Betrag von X und Y in die durch die Vorzeichen bestimmte Richtung gefahren. Ist IP=0,so bleibt der Stift gehoben und ist IP>0, so wird der Stift gesenkt und somit eine Linie gezogen.

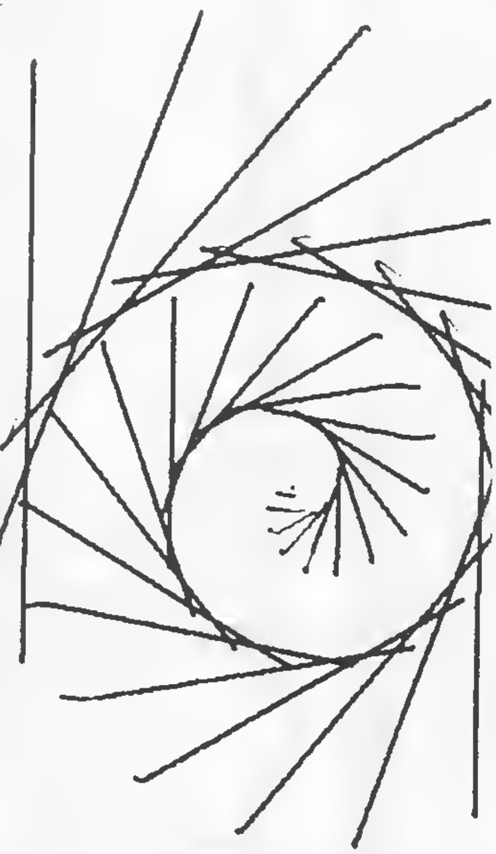
CALL LINK< "PLTA",X,Y,IP)
Dieses Programm zeichnet absolut, d.h., der Stift bewegt sich vom Nullpunkt aus gesehen auf den Punkt mit den Koordinaten X und Y. Der Nullpunkt wird durch die Position festgelegt, in der der Stift vor dem ersten Aufruf von "PLTA" ist. Für IP gibt es drei Möglichkeiten:

1. IP<: setze diesen Punkt als Nullpunkt.
2. IP=0: fahre zu der angegebenen Position mit gehobenem Stift.
3. IP>0: fahre zu der angegebenen Position mit gesenktem Stift (Linie)

Zur leichteren Handhabung des Plotters dienen die beiden folgenden Programme:

CALL LINK<("RESET")
stellt die Startwerte für alle Plotprogramme her: Die augenblickliche Position ist der Nullpunkt, und der Stift ist gehoben. Die Skalierung für die X- und Y-Richtung wird zu 57,14 Motorschritte pro 1,0 cm gesetzt.

CALL LINK<("FAKTOR",XFAK, YFAK)
Hiermit werden die Skalierungsfaktoren für die X- und Y-Richtung vorgegeben. Es wird definiert, wieviele Motorschritte für X=1 bzw. Y=1 zugeordnet werden sollen.
Da ein Motorschritt eine viertel Umdrehung der Gewindestangen bewirkt und ein M5-Gewinde eine Steigung von 0,7 mm/Umdrehung hat, bewegt sich der Stift bei einem Motorschritt um 0,7/4,0 mm = 0,175 mm. Daraus folgt, wenn die X- und Y-Koordinaten in cm eingegeben werden sollen, muß XFAK und YFAK = 10/0,175 = 57,14 gesetzt werden. Mit diesem Faktor wird also erreicht, daß X und Y in "PLTR" und "PLTA" in cm eingegeben werden können. Durch den Aufruf von "RESET" wird dieser Faktor auch gesetzt.



Anmerkung: Das Mischen von "PLTA" und "PLTR" ohne "RESET" ist nicht erlaubt!
Da es sich hier um Routinen handelt, die in Assembler geschrieben sind, gibt es zwei Möglichkeiten, sie in den Rechner zu laden.
1. Den Quellcode zu übersetzen und zu laden. Dafür ist aber dann eine Diskettenstation notwendig.
2. Die Pokeliste von Kassette mit Hilfe meines beigefügten BASIC-Ladeprogramms in das Low-Memory poken. Das Plotprogramm belegt dann den Speicher von> 3000 bis >34BE.

Matthias Wahne

```
        AORG >3000
        DEF  PLTR
        DEF  PLTA
        DEF  RESET
        DEF  FAKTOR
CRU     EQU  >1600
NUMREF  EQU  >200C
XMLLNK  EQU  >2018
FAC     EQU  >834A
ARG     EQU  >835C
RTN     EQU  >0070
STATUS  EQU  >837C
GPL     EQU  >83E0
FDIV    EQU  >0FF4
FADD    EQU  >0D80
FMUL    EQU  >0E88
MY      BSS  32
SR11    BSS  2
RT1     BSS  2
RT2     BSS  2
TAST    EQU  >8374
KSCN    EQU  >201C
ZW      BSS  2
IX      BSS  2
CDE     BSS  2
CDEA    BSS  2
CDXY    BSS  2
IY      BSS  2
IPEN    BSS  2
X       BSS  8
Y       BSS  8
S       DATA >0000
        DATA >0000
        DATA >0000
        DATA >0000
G       BSS  8
XFAK    DATA >4039
        DATA >0E1C
        DATA >0000
        DATA >0000
YFAK    DATA >4039
        DATA >0E1C
        DATA >0000
        DATA >0000
XA      DATA >0000
        DATA >0000
        DATA >0000
        DATA >0000
YA      DATA >0000
        DATA >0000
        DATA >0000
        DATA >0000
P5      DATA >C0CE
        DATA >0000
        DATA >0000
        DATA >0000
EI      DATA >4001
        DATA >0000
        DATA >0000
```

LISTING

```
        DATA >ØØØØ
XUB     DATA >1EØØ
        DATA >12ØØ
        DATA >1CØØ
        DATA >1AØØ
        DATA >ØØØØ
        DATA >16ØØ
        DATA >14ØØ
        DATA >1ØØØ
        DATA >18ØØ
NULL    DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
RFAKT   DATA >4Ø39
        DATA >ØE1C
        DATA >ØØØØ
        DATA >ØØØØ
PAXY    DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
        DATA >ØØØØ
TRD     MOV  *2+,*3+
        MOV  *2+,*3+
        MOV  *2+,*3+
        MOV  *2,*3
        RT
XTRP    MOV  11,@RT2
        MOV  3,@ZW
        LI   3,FAC
        BL   @TRD
        MOV  @ZW,2
        LI   3,ARG
        BL   @TRD
        MOV  @RT2,11
        RT
XTRG    MOV  11,@RT2
        LI   2,FAC
        MOV  @ZW,3
        BL   @TRD
        MOV  @RT2,11
        RT
MYADD   MOV  11,@RT1
        BL   @XTRP
        LWPI GPL
        BL   @FADD
        LWPI MY
        MOV  @RT1,11
        RT
MYMUL   MOV  11,@RT1
        BL   @XTRP
        LWPI GPL
        BL   @FMUL
        LWPI MY
        MOV  @RT1,11
        RT
MYDIV   MOV  11,@RT1
        BL   @XTRP
        LWPI GPL
        BL   @FDIV
        LWPI MY
        MOV  @RT1,11
        RT
BASIC   CLR  Ø
        MOVB Ø,@STATUS
        LWPI GPL
        B    @RTN
PARP    MOV  11,@SR11
        LI   8,>ØØØ8
        LI   9,>ØØØ2
PN      CLR  Ø
        MOV  9,1
        BLWP @NUMREF
        LI   3,X
        A    8,3
        LI   2,FAC
        BL   @TRD
        LI   2,XFAK
        A    8,2
        LI   3,X
        A    8,3
        BL   @MYMUL
        BL   @XTRG
        LI   3,X
        A    8,3
        LI   2,XA
        A    8,2
        BL   @MYADD
        BL   @XTRG
        BLWP @XMLLNK
        DATA >12B8
        MOV  @FAC,@IX(8)
        LI   3,XA
        A    8,3
        LI   2,NULL
        BL   @TRD
        CLR  8
        DEC  9
        JNE  PN
        CLR  Ø
        LI   1,>ØØØ3
        BLWP @NUMREF
        BLWP @XMLLNK
        DATA >12B8
        MOV  @FAC,@IPEN
        MOV  @SR11,11
        RT
PSTA    MOV  11,@SR11
        LI   12,CRU
GG      CLR  4
        MOVB 4,@TAST
        MOVB 4,@STATUS
        BLWP @KSCN
        MOVB @STATUS,4
        JNE  GK
        LI   4,>22ØØ
GO      DEC  4
        JNE  GO
        STCR 4,8
        JNE  GG
GK      MOV  @SR11,11
        RT
POUT    MOV  11,@SR11
        CLR  4
        LI   12,CRU
        LDCR 4,8
        MOV  @CDE,4
        LDCR 4,8
        ORI  4,>Ø1ØØ
        LDCR 4,8
        MOV  @SR11,11
        RT
PLTR    LWPI MY
        BL   @PARP
PLT     LI   5,>ØAØØ
        CLR  Ø
        C    @IPEN,Ø
        JEQ  P1
        LI   5,>ØCØØ
P1      MOV  5,@CDE
        BL   @PSTA
        BL   @POUT
        CLR  6
        CLR  7
        ABS  @IX
        JEQ  XO
        JGT  GX
        LI   6,>FFFF
        B    @XO
GX      LI   6,>ØØØ1
XO      ABS  @IY
        JEQ  YO
        JGT  GY
        LI   7,>FFFF
        B    @YO
GY      LI   7,>ØØØ1
YO      MOV  6,5
        A    5,5
        A    6,5
        A    7,5
        AI   5,>ØØØ4
        A    5,5
        MOV  @XUB(5),@CDEA
        CLR  8
        CLR  9
        C    @IX,@IY
        JGT  XG
        MOV  7,5
        MOV  @IY,6
        AI   5,>ØØØ4
        A    5,5
        MOV  @XUB(5),@CDXY
        LI   8,>ØØØ8
        CLR  9
        LI   2,Y
        LI   3,X
```

LISTING

```
CON     BL    @MYDIV
        LI    2,FAC
        LI    3,G
        BL    @TRD
        ABS   @G
        JMP   DO
XG      MOV   6,5
        A     5,5
        A     6,5
        AI    5,>ØØØ4
        A     5,5
        MOV   @XUB(5),@CDXY
        CLR   8
        LI    9,>ØØØ8
        MOV   @IX,6
        LI    2,X
        LI    3,Y
        JMP   CON
DO      INC   6
        LI    2,NULL
        LI    3,S
        BL    @TRD
GOON    DEC   6
        JLE   ENDE
        MOV   @CDXY,@CDE
        LI    3,XA
        A     8,3
        LI    2,EI
        BL    @MYADD
        BL    @XTRG
        LI    3,S
        LI    2,G
        BL    @MYADD
        BL    @XTRG
        LI    2,S
        LI    3,P5
        BL    @MYADD
        MOV   @FAC,5
        JLT   GOPLT
        LI    3,XA
        A     9,3
        LI    2,EI
        BL    @MYADD
        BL    @XTRG
        LI    3,S
        NEG   @EI
        LI    2,EI
        BL    @MYADD
        BL    @XTRG
        NEG   @EI
        MOV   @CDEA,@CDE
GOPLT   BL    @PSTA
        MOV   4,4
        JNE   ENDE
        BL    @POUT
        JMP   GOON
ENDE    MOV   @X,8
        JLT   XX
        JEQ   XX
        NEG   @XA
XX      LI    3,XA
        LI    2,X
        BL    @MYADD
        BL    @XTRG
        MOV   @Y,9
        JLT   YY
        JEQ   YY
        NEG   @YA
YY      LI    3,YA
        LI    2,Y
        BL    @MYADD
        BL    @XTRG
        B     @BASIC
PLTA    LWPI  MY
        BL    @PARP
        LI    8,>ØØØ8
        LI    9,>ØØØ2
AA      NEG   @PAXY(8)
        LI    2,PAXY
        A     8,2
        LI    3,X
        A     8,3
        BL    @MYADD
        LI    2,X
        A     8,2
        LI    3,PAXY
        A     8,3
        BL    @TRD
        LI    2,FAC
        LI    3,X
        A     8,3
        BL    @MYADD
        LI    2,X
        A     8,2
        LI    3,PAXY
        A     8,3
        BL    @TRD
        LI    2,FAC
        LI    3,X
        A     8,3
        BL    @TRD
        BLWP  @XMLLNK
        DATA  >12B8
        MOV   @FAC,@IX(8)
        CLR   8
        DEC   9
        JNE   AA
        MOV   @IPEN,9
        JGT   APLT
        JEQ   APLT
        B     @BASIC
APLT    B     @PLT
FAKTOR  LWPI  MY
        LI    9,2
        LI    8,8
PF      CLR   Ø
        MOV   9,1
        BLWP  @NUMREF
        LI    2,FAC
        LI    3,XFAC
        A     8,3
        BL    @TRD
        CLR   8
        DEC   9
        JNE   PF
        B     @BASIC
RESET   LWPI  MY
        LI    4,>ØAØØ
        MOV   4,@CDE
        BL    @POUT
        LI    4,>1EØØ
        MOV   4,@CDE
        BL    @POUT
        LI    4,>18ØØ
        MOV   4,@CDE
        BL    @PSTA
        BL    @POUT
        LI    2,NULL
        LI    3,S
        BL    @TRD
        LI    4,2
        LI    5,8
        LI    2,NULL
        LI    3,XA
        A     5,3
        BL    @TRD
        LI    2,NULL
        LI    3,PAXY
        A     5,3
        BL    @TRD
        LI    2,RFACT
        LI    3,XFAC
        A     5,3
        BL    @TRD
        CLR   5
        DEC   4
        JNE   RR
        B     @BASIC
        END
```

LISTINGS

```
10 !************************
20 !* Textplot Subroutinen *
30 !* Copyright  M. WAHNER *
40 !* CALL PLTXT("str",X,Y,*
50 !* size,grad)          *
60 !* CALL ROTATE(X,Y,grad)*
70 !* Benltigte Ger(te :   *
80 !* Konsole & Ex-Basic   *
90 !* Selbstbau - Plotter  *
95 !* Plotsoftware f}r den *
96 !* Plotter  (M. WAHNER) *
99 !************************
100 CALL LINK("RESET")
110 CALL PLTXT("EIN DRUCKER
KANN",-3.0,6,.05,0)
120 CALL PLTXT("ZWAR WESENTL
ICH",-2.75,5,.05,0)
130 CALL PLTXT("SCHNELLER SC
HREIBEN",-3.75,4,.05,0)
140 A$=" ABER DAS KANN ER NI
CHT!"
150 X=0.0 :: Y=1.50 :: R=0 :
: FOR I=1 TO 24 :: XX=X :: Y
Y=Y :: CALL ROTATE(XX,YY
,R):: CALL PLTXT(SEG$(A$,I,1
),XX,YY,.05,R):: R=R-15 :: N
EXT I
160 CALL LINK("PLTA",-4,-4,0
):: CALL LINK("RESET")
170 X=1 :: Y=0.0 :: R=0 :: F
OR I=1 TO 8 :: XX=X :: YY=Y
:: CALL ROTATE(XX,YY,R):
: CALL PLTXT("PLOT",XX,YY,.0
5,R):: R=R+45 :: NEXT I
180 CALL LINK("PLTA",7,0,0):
: CALL LINK("RESET")
190 S=.25 :: ST=0 ! RELATIV
PLOTTEN MEANDER
200 FOR I=1 TO 5 :: ST=ST+S
:: CALL LINK("PLTR",ST,0,1):
: CALL LINK("PLTR",0,ST,
1):: ST=ST+S :: CALL LINK("P
LTR",-ST,0,1):: CALL LINK("P
LTR",0,-ST,1):: NEXT I
210 CALL LINK("RESET"):: END
30000 ! Data fuer Buchstaben
30012 DATA 1,0,0,0
30013 DATA 4,3,0,0,3,1,1,3,3
,0,3,8,1
30014 DATA 1,0,0,0
30015 DATA 1,0,0,0
30016 DATA 1,0,0,0
30017 DATA 1,0,0,0
30018 DATA 1,0,0,0
30019 DATA 1,0,0,0
30020 DATA 1,0,0,0
30021 DATA 1,0,0,0
30022 DATA 1,0,0,0
30023 DATA 1,0,0,0
30024 DATA 1,0,0,0
30025 DATA 2,2,4,0,5,4,1
30026 DATA 2,3,0,0,3,0,1
30027 DATA 2,1,0,0,5,8,1
30028 DATA 5,1,0,0,1,8,1,5,8
,1,5,0,1,1,0,1
30029 DATA 3,0,5,0,3,8,1,3,0
,1
30030 DATA 6,0,6,0,0,8,1,6,8
,1,6,5,1,0,0,1,6,0,1
30031 DATA 7,0,6,0,2,8,1,6,6
,1,3,4,1,6,2,1,3,0,1,0,2,1
30032 DATA 5,0,8,0,0,3,1,6,3
,1,3,6,0,3,0,1
30033 DATA 6,3,0,1,6,3,1,3,5
,1,0,5,1,0,8,1,6,8,1
30034 DATA 6,0,4,0,6,4,1,6,0
,1,0,0,1,0,8,1,6,8,1
30035 DATA 3,2,0,0,6,8,1,0,8
,1
30036 DATA 7,0,4,0,6,4,1,6,0
,1,0,0,1,0,8,1,6,8,1,6,3,1
30037 DATA 5,6,0,1,6,8,1,0,8
,1,0,4,1,6,4,1
30038 DATA 4,3,5,0,3,5,1,3,2
,0,3,2,1
30039 DATA 4,3,5,0,3,5,1,3,2
,0,3,0,1
30040 DATA 1,0,0,0
30041 DATA 1,0,0,0
30042 DATA 1,0,0,0
30043 DATA 1,0,0,0
30044 DATA 1,0,0,0
30045 DATA 4,3,8,1,6,0,1,1.1
,3,0,4.9,3,1
30046 DATA 9,0,8,1,3,8,1,6,6
,1,3,4,1,0,4,1,4,4,1,6,2,1,4
,0,1,0,0,1
30047 DATA 4,6,8,0,0,8,1,0,0
,1,6,0,1
30048 DATA 6,0,8,1,4,8,1,6,6
,1,6,2,1,4,0,1,0,0,1
30049 DATA 6,0,8,1,6,8,1,4,4
,0,0,4,1,0,0,0,6,0,1
30050 DATA 4,0,8,1,6,8,1,4,4
,0,0,4,1
30051 DATA 6,6,8,0,0,8,1,0,0
,1,5,0,1,5,4,1,3,4,1
30052 DATA 5,0,8,1,0,4,0,6,4
,1,6,8,0,6,0,1
30053 DATA 2,4,8,0,4,0,1
30054 DATA 5,1,1,0,1,0,1,4,0
,1,4,8,1,3,8,1
30055 DATA 5,0,8,1,6,8,0,0,3
,1,1,4,1,5,0,1
30056 DATA 3,0,8,0,0,0,1,6,0
,1
30057 DATA 4,0,8,1,3,4,1,6,8
,1,6,0,1
30058 DATA 3,0,8,1,6,0,1,6,8
,1
```

LISTINGS

```
30059 DATA 4,0,8,1,6,8,1,6,0
,1,0,0,1
30060 DATA 4,0,8,1,6,8,1,6,4
,1,0,4,1
30061 DATA 7,0,8,1,6,8,1,6,3
,1,4,0,1,0,0,1,4,3,0,6,0,1
30062 DATA 5,0,8,1,6,8,1,6,4
,1,0,4,1,6,0,1
30063 DATA 5,6,0,1,6,2,1,0,6
,1,0,8,1,6,8,1
30064 DATA 4,3,0,0,3,8,1,0,8
,0,6,8,1
30065 DATA 4,0,8,0,0,0,1,6,0
,1,6,8,1
30066 DATA 3,0,8,0,3,0,1,6,8
,1
30067 DATA 5,0,8,0,0,0,1,3,4
,1,6,0,1,6,8,1
30068 DATA 3,6,8,1,0,8,0,6,0
,1
30069 DATA 5,0,8,0,3,4,1,6,8
,0,3,4,1,3,0,1
30070 DATA 4,0,8,0,6,8,1,0,0
,1,6,0,1
30071 DATA 1,0,0,0
30072 SUB PLTXT(TX$,X,Y,TFAK
,ROT)
30073 L=LEN(TX$):: IF L<=0 T
HEN SUBEXIT
30074 CALL LINK("PLTA",X,Y,0
)
30075 FOR I=1 TO L :: N=ASC(
SEG$(TX$,I,1))
30076 IF N<32 OR(N>90 AND N<
97)OR N>122 THEN 30144
30077 IF N>96 THEN N=N-32
30078 N=N-31
30079 IF N<=20 THEN GOTO 300
80 ELSE IF N>40 THEN GOTO 30
082 ELSE 30081
30080 ON N GOTO 30083,30084,
30085,30086,30087,30088,3008
9,30090,30091,30092,3009
3,30094,30095,30096,30097,30
098,30099,30100,30101,30102
30081 ON N-20 GOTO 30103,301
04,30105,30106,30107,30108,3
0109,30110,30111,30112,3
0113,30114,30115,30116,30117
,30118,30119,30120,30121,301
22
30082 ON N-40 GOTO 30123,301
24,30125,30126,30127,30128,3
0129,30130,30131,30132,3
0133,30134,30135,30136,30137
,30138,30139,30140,30141,301
42
30083 RESTORE 30012 :: GOTO
30143
30084 RESTORE 30013 :: GOTO
30143
30085 RESTORE 30014 :: GOTO
30143
30086 RESTORE 30015 :: GOTO
30143
30087 RESTORE 30016 :: GOTO
30143
30088 RESTORE 30017 :: GOTO
30143
30089 RESTORE 30018 :: GOTO
30143
30090 RESTORE 30019 :: GOTO
30143
30091 RESTORE 30020 :: GOTO
30143
30092 RESTORE 30021 :: GOTO
30143
30093 RESTORE 30022 :: GOTO
30143
30094 RESTORE 30023 :: GOTO
30143
30095 RESTORE 30024 :: GOTO
30143
30096 RESTORE 30025 :: GOTO
30143
30097 RESTORE 30026 :: GOTO
30143
30098 RESTORE 30027 :: GOTO
30143
30099 RESTORE 30028 :: GOTO
30143
30100 RESTORE 30029 :: GOTO
30143
30101 RESTORE 30030 :: GOTO
30143
30102 RESTORE 30031 :: GOTO
30143
30103 RESTORE 30032 :: GOTO
30143
30104 RESTORE 30033 :: GOTO
30143
30105 RESTORE 30034 :: GOTO
30143
30106 RESTORE 30035 :: GOTO
30143
30107 RESTORE 30036 :: GOTO
30143
30108 RESTORE 30037 :: GOTO
30143
30109 RESTORE 30038 :: GOTO
30143
30110 RESTORE 30039 :: GOTO
30143
30111 RESTORE 30040 :: GOTO
30143
30112 RESTORE 30041 :: GOTO
30143
30113 RESTORE 30042 :: GOTO
30143
30114 RESTORE 30043 :: GOTO
```

```
30143
30115 RESTORE 30044 :: GOTO
30143
30116 RESTORE 30045 :: GOTO
30143
30117 RESTORE 30046 :: GOTO
30143
30118 RESTORE 30047 :: GOTO
30143
30119 RESTORE 30048 :: GOTO
30143
30120 RESTORE 30049 :: GOTO
30143
30121 RESTORE 30050 :: GOTO
30143
30122 RESTORE 30051 :: GOTO
30143
30123 RESTORE 30052 :: GOTO
30143
30124 RESTORE 30053 :: GOTO
30143
30125 RESTORE 30054 :: GOTO
30143
30126 RESTORE 30055 :: GOTO
30143
30127 RESTORE 30056 :: GOTO
30143
30128 RESTORE 30057 :: GOTO
30143
30129 RESTORE 30058 :: GOTO
30143
30130 RESTORE 30059 :: GOTO
30143
30131 RESTORE 30060 :: GOTO
30143
30132 RESTORE 30061 :: GOTO
30143
30133 RESTORE 30062 :: GOTO
30143
30134 RESTORE 30063 :: GOTO
30143
30135 RESTORE 30064 :: GOTO
30143
30136 RESTORE 30065 :: GOTO
30143
30137 RESTORE 30066 :: GOTO
30143
30138 RESTORE 30067 :: GOTO
30143
30139 RESTORE 30068 :: GOTO
30143
30140 RESTORE 30069 :: GOTO
30143
30141 RESTORE 30070 :: GOTO
30143
30142 RESTORE 30071 :: GOTO
30143
30143 CALL TX(X,Y,TFAK,ROT)
30144 NEXT I
30145 SUBEND
```



```
30146 SUB TX(XA,YA,TF,ROT)::
 READ M :: YN=0 :: XN=8 :: C
ALL ROTATE(XN,YN,ROT)
30148 FOR I=1 TO M :: READ X
,Y,IP :: CALL ROTATE(X,Y,ROT
):: CALL LINK("PLTA",X*T
F+XA,Y*TF+YA,IP):: NEXT I ::
 XA=XA+XN*TF :: YA=YA+YN*TF
:: CALL LINK("PLTA",XA,Y
A,0):: SUBEND
30149 SUB ROTATE(X,Y,R):: CR
=COS(PI/180*R):: SI=SIN(PI/1
80*R):: XR=X*CR-Y*SI ::
YR=X*SI+Y*CR :: X=XR :: Y=YR
 :: SUBEND
30155 REM Ende des Programms
```

TIP ZUM MODULPORT

Auch ich hatte Probleme mit dem Modul Extended Basic II plus. Nach mehreren bösen Abstürzen entschloß ich mich zu einer Reinigung der Kontakte am Modulsockel, mit der ich schon einmal Erfolg hatte im Zusammenhang mit dem originalen XB-Modul. Aber diesmal verschwanden die Abstürze nicht, im Gegenteil: Plötzlich machte der Computer völlig grundlos einen Sprung im Programm, der auch nach einem Neustart blieb.
Ich schloß auf einen Fehler im Modul, da die Programme mit dem Original-Modul einwandfrei liefen, und tauschte es um. Die Sprünge verschwanden, die Abstürze nicht. Zeitweise stürzte der Computer schon beim Anwählen in der Hauptwahlliste ab. Von einem Bekannten kaufte ich dann eine zweite Konsole, aber auch hier entstanden die Abstürze. Wieder griff ich zum Schraubendreher und öffnete gleich beide Konsolen. Beim Herumpröbeln stellte es sich heraus, daß die Modulsockel ausgeleiert waren, allerdings nicht je gleich stark, und auch der Modulschacht der Konsole zeigte Abnützungserscheinungen. Darüber hinaus sind die Leiterbahnen auf der Platine des Extended Basic II plus breiter als beim originalen XB-Modul. Ich nahm an, daß durch diese Faktoren beim Einschieben des Moduls ein falscher Kontakt entstehen kann. Beim Zusammenbauen der Konsolen tauschte ich die beiden Modulsockel gegenseitig aus, ein Versuch, und siehe da: Die Probleme tauchten nicht mehr auf. Dieses Frühjahr nun kaufte ich von einem anderen Bekannten eine ganze Anlage. Bei dieser dritten Konsole kamen die Abstürze wieder. Hier ging ich aber anders vor: Ich bestellte bei Texas Instruments (Schweiz) zwei andere Modulsockel, die schon drei Tage später eintrafen, zu einem Preis von SFr. 10,–/Stück. Jetzt ärgert mich kein Absturz mehr, das Arbeiten ist zum Genuß geworden. *Josef Kronenberg*

APFEL-MÄNNCHEN

Dieses Programm stellt einen Beitrag zum Thema Computerkunst dar: Es erzeugt sogenannte Apfelmännchen. Dies sind Gebilde, die durch wiederholte Anwendung einer einfachen mathematischen Formel auf komplexe Zahlen entstehen.
Komplexe Zahlen bestehen aus einem Real- und einem Imaginärteil und haben allgemein die Form X+Y*i. Dabei stellt X den Realteil und Y den Imaginärteil dar, i dient der Erkennung des Imaginärteils und ist mit SQR(−1) gleichzusetzen. Da X und Y Realzahlen sind, bilden die komplexen Zahlen ein zweidimensionales Koordinatensystem. Man spricht deshalb auch von der Ebene der komplexen Zahlen oder der komplexen Ebene, wobei der Absolutwert gleich dem Abstand zum Ursprung ist und sich nach dem Satz des Pythagoras aus SQR(X*X+Y*Y) berechnet. Beim Rechnen mit komplexen Zahlen gelten die gleichen Regeln wie beim Rechnen mit Summen (von Realzahlen).
Wenn man nun die Zahl K=0+0i wiederholt in die Folge Kneu=Kalt^2+C einsetzt, so hängt ihr Verhalten stark von dem Wert von C ab. Ist der Absolutwert von C größer als 2, so entfernen sich die Folgeglieder recht schnell vom Ursprung. Bei Zah-



len mit einem Absolutwert von weniger als 2 irrt die Folge jedoch eine Weile in diesem Bereich herum, bevor sie das „Gefängnis“ verlassen kann. Dazu können mehr als 1000 Folgeglieder vonnöten sein. Wenn man die Anzahl der Iterationen als Farbe auffaßt und die komplexe Ebene solchermaßen einfärbt, so ergibt sich in der Umgebung des Ursprungs eine Figur, die man ihrer Form wegen „Apfelmännchen“ getauft hat. Das besondere an dieser Figur sind ihre Ränder: Man kann sie nämlich beliebig vergrößern, ohne eine kleinste Form zu entdecken. Jede Form wiederholt sich in irgendeiner Weise in irgendeiner Vergrößerung, wobei keine Wiederholung völlig identisch zur anderen ist.
So zeigen sich zum Beispiel ganze Apfelmännchen mit kleinen Abweichungen in manchen Bereichen des Randes. Doch auch Spiralen, Seepferdchen und andere Muster lassen sich dort finden. Das heißt aber nicht, daß die Erzeugung solcher Muster, auch „Fractals“ genannt, reine Spielerei wären.
Im Gegenteil: Wie man vor einiger Zeit festgestellt hat, treten solche Muster auch in der Natur auf. Nämlich bei der Änderung von magnetischen Zuständen. Dabei bilden sich an den Rändern von Magnetfeldzonen ähnliche Erscheinungen; möglicherweise ein Zeichen dafür, daß es die vielgesuchten kleinsten Teilchen in der Materie nicht gibt. Doch gleichgültig von welchem Standpunkt man die Fractals betrachtet, schön sind sie allemal.
Das Programm Fractal wird unter TI-Basic wie folgt aufgerufen: Zunächst muß mit CALL FILES(1) und NEW der Speicherplatz für den Disccontroller begrenzt werden. Dann wird mit CALL LOAD das Vorprogramm für das Abspeichern des fertigen Bildes geladen. Danach muß noch das Hauptprogramm mittels CALL LOAD geladen werden. Zum Start des Programms gibt man CALL LINK(“START“, Xvon, Xbis, Yvon, Ybis, Anzahl) ein. Dabei sind Xvon, Xbis, Yvon,Ybis die Feldgrenzen für den Bereich der komplexen Zahlen, der berechnet werden soll, Anzahl ist die Anzahl der Iterationen, die maximal durchgeführt werden sollen. Um das gesamte Apfelmännchen zu berechnen, sind als Parameter −0.7,2.3,−1.25,1.25, 50 geeignet. Je kleiner der gewählte Bereich ist, desto größer muß man auch die Anzahl der Iterationen wählen.
Das Programm arbeitet mit dem BITMAP-MODE in Schwarz/Weiß, um die größtmögliche Auflösung zu nutzen. Die Anzahl der zum Verlassen des Ursprungsbereichs benötigten Iterationen werden deshalb nicht als Farbe interpretiert, sondern nach folgendem Schlüssel aufgeteilt: Ungerade Anzahl – weiß, gerade Anzahl – schwarz. Die Routine zum Abspeichern des fertigen Bildes kann beliebig ersetzt werden, falls eine andere Art der Speicherung erwünscht ist. Dabei muß lediglich beachtet werden, daß die Routine mit BLWP aufgerufen wird, und der Einsprungvektor mit dem Label SAVE DEFiniert werden muß. Die beigefügte Routine speichert nach Fertigstellung des Bildes und Drükken der ENTER-Taste das Bild auf DSK1 mit dem Namen MANDELBROT ab. Zum Laden eines Bildes von Diskette ist der Loader (auf der Diskette DSKLOADO genannt) mittels CALL LOAD zu laden und mit CALL LINK(“LOAD“,“Filename“) zu starten. Auch dabei muß zunächst CALL FILES(1) und NEW eingegeben werden.
Das Programm ist mit dem EDITOR ASSEMBLER wie mit dem MINIMEMORY lauffähig. Wenn das EDITOR ASSEMBLER-Modul verwendet wird, muß allerdings das am Anfang stehende AORG-Statement (in allen drei Teilprogrammen) gelöscht und das Programm neu assembliert werden. Bei Verwendung des MINIMEMORY-Moduls ergibt sich jedoch der Vorteil, daß die Peripheriebox während der Berechnung abgeschaltet werden kann, was bei der langen Dauer der Berechnung (zwischen 6 und 48 Stunden, je nach Iterationsanzahl) einige Lüfterumdrehungen spart und ein Übernachtberechnen ermöglicht.

LISTINGS

```
*  *********************************
* ***********************************
* *                                 *
* *          MANDELBROTMENGE        *
* *               BY                *
* *          KARSTEN KUTOWSKI       *
* *                                 *
* * LADEN: -CALL FILES(1)           *
* *        -NEW                     *
* *        -CALL LOAD("DSKn.DSKSAVEO")*
* *        -CALL LOAD("DSKn.FRACTALO")*
* *                                 *
* * STARTEN:                        *
* * -CALL LINK("START",XA,XE,YA,YE,I) *
* *                                 *
* * XA = ANFANG REALTEIL            *
* * XE = ENDE REALTEIL              *
* * YA = ANFANG IMAGINAERTEIL       *
* * YE = ENDE IMAGINAERTEIL         *
* * I  = ANZAHL DER ITERATIONEN     *
* *                                 *
* *         1.TEIL: FRACTAL         *
* * ASSEMBLIERT: 'FRACTALO'         *
* *                                 *
* ***********************************
*  *********************************

       AORG >7118   ERSTE FREIE ADRESSE IM MINIMEM-RAM

VWTR   EQU >6034    UTILITY-EQUATES
VSBW   EQU >6024
VSBR   EQU >602C
XMLLNK EQU >601C
NUMREF EQU >6044
KSCAN  EQU >6020

FAC    EQU >834A    FLOATING POINT ACKUMULATOR
ARG    EQU >835C    ZWEITES ARGUMENT BEI BERECHNUNGEN
FADD   EQU >0600    ADDITION
FSUB   EQU >0700    SUBTRAKTION
FMUL   EQU >0800    MULTIPLIKATION
FDIV   EQU >0900    DIVISION
FCOM   EQU >0A00    VERGLEICH
CFI    EQU >1200    FLOATING POINT->INTEGER
CIF    EQU >7200    INTEGER->FLOATING POINT (REF CIF BEI EDITOR ASSEMBLER)

       REF SAVE     KANN ENTFALLEN WENN ABSPEICHERN NICHT ERWUENSCHT

       DEF START

RDATA  DATA >0002,>01E0,>0206,>03FF     DATEN FUER WRITE ONLY REGISTER
       DATA >0403,>0536,>0607,>07F1     IM BITMAP MODE

D2     DATA 2
D191   DATA 191         BILDSCHIRMGRENZEN
D255   DATA 255

KX     BSS 2
```

LISTINGS

```
R1ØØ    BSS 8           FLOATING POINT VARIABLEN
REAL2   BSS 8
XU      BSS 8
XO      BSS 8
YU      BSS 8
YO      BSS 8
DX      BSS 8
DY      BSS 8
XC      BSS 8
YC      BSS 8
M       BSS 8
N       BSS 8
XZ      BSS 8
YZ      BSS 8
XX      BSS 8
YY      BSS 8


BITMAP DATA WS,BITMA1   BLWP-VEKTOR FUER EINSCHALTEN DES BITMAP MODES

PLOT    DATA WS,PLOT1   BLWP-VEKTOR FUER PLOT-ROUTINE

WS      BSS 32          BLWP-WORKSPACE

GETP    MOV 11,1Ø       HOLT NUMERISCHEN PARAMETER
        CLR Ø
        BLWP @NUMREF
        MOV 2,Ø
        BL @FFAC
        B *1Ø

FFAC    LI 1,FAC        COPIERT REALWERT AUS DEM FAC NACH *RØ
        MOV *1+,*Ø+
        MOV *1+,*Ø+
        MOV *1+,*Ø+
        MOV *1+,*Ø+
        RT

TFAC    LI 1,FAC        COPIERT REALWERT VON *RØ ZUM FAC
        MOV *Ø+,*1+
        MOV *Ø+,*1+
        MOV *Ø+,*1+
        MOV *Ø+,*1+
        RT

FARG    LI 1,ARG        COPIERT REALWERT AUS DEM ARG NACH *RØ
        MOV *1+,*Ø+
        MOV *1+,*Ø+
        MOV *1+,*Ø+
        MOV *1+,*Ø+
        RT

TARG    LI 1,ARG        COPIERT REALWERT VON *RØ NACH ARG
        MOV *Ø+,*1+
        MOV *Ø+,*1+
        MOV *Ø+,*1+
        MOV *Ø+,*1+
        RT
```

LISTINGS

```
BITMA1 LI 1,RDATA             SCHALTET BITMAP MODE EIN
       MOVB @3(1),@>83D4 COPIE DES INHALTS VON VDP R1
BITMA2 MOV *1+,0
       BLWP @VWTR             VDP REGISTER LADEN
       CI 1,RDATA+16
       JNE BITMA2
       CLR 0                  PATTERN DESCRIPTOR INITIALISIEREN
       CLR 1
BITMA3 BLWP @VSBW
       INC 0
       CI 0,>1800
       JNE BITMA3
BITMA4 BLWP @VSBW             SCREEN IMAGE TABLE INITIALISIEREN
       AI 1,>0100
       INC 0
       CI 0,>1B00
       JNE BITMA4
       LI 1,>D000             SPRITE ATTRIBUTE LIST: KEINE SPRITES
       BLWP @VSBW
       LI 0,>2000             COLOR TABLE INITIALISIEREN
       LI 1,>F100             FARBEN FG:WEISS HG:SCHWARZ
BITMA5 BLWP @VSBW
       INC 0
       CI 0,>3800
       JNE BITMA5
       RTWP

PLOT1  MOV *13+,0             SETZEN EINES PIXELS R0:YWERT R1:XWERT
       MOV *13+,1                                 R2:GERADE->NICHTS TUN
       MOV *13,2                                     UNGERADE->PIXEL SETZEN
       AI 13,-4
       ANDI 2,1               IST R2 GERADE?
       JEQ PLOT2              WENN JA ABHAUEN
       MOV 1,4                BIT-&BYTEOFFSET BERECHNEN
       SLA 4,5
       SOC 1,4
       ANDI 4,>FF07
       MOV 0,5
       ANDI 5,7
       A 0,4
       S 5,4
       LI 3,>8000
       MOV 5,0
       SRC 3,0                BIT JUSTIEREN
       MOV 4,0
       CLR 1
       BLWP @VSBR             BYTE AUS PATTERN DESCRIPTOR LESEN
       SOC 3,1                MIT JUSTIERTEM BIT OR-VERKNUEPFEN
       BLWP @VSBW             UND WIEDER SCHREIBEN
PLOT2  RTWP

START  LI 2,XU                HAUPTPROGRAMM
       LI 1,1                 PARAMETER HOLEN
       BL @GETP
       LI 2,XO
       LI 1,2
       BL @GETP
       LI 2,YU
```

```
        LI 1,3
        BL @GETP
        LI 2,YO
        LI 1,4
        BL @GETP
        LI 2,ARG
        LI 1,5
        BL @GETP
        BLWP @XMLLNK        UMWANDELN DES LETZTEN PARAMETERS IN INTEGER
        DATA CFI
        MOV @FAC,@KX
        BLWP @BITMAP        BITMAP-MODE EINSCHALTEN
        MOV @KX,7
        INC 7
        LI Ø,1ØØ            1ØØ -> REAL 1ØØ
        MOV Ø,@FAC
        BLWP @XMLLNK
        DATA CIF
        LI Ø,R1ØØ
        BL @FFAC
        LI Ø,XO             DX=(XO-XU)/255 X-ABSTAND ZWISCHEN ZWEI PUNKTEN
        BL @TARG                           (BEZUEGLICH DES BILDSCHIRMS)
        LI Ø,XU
        BL @TFAC
        BLWP @XMLLNK
        DATA FSUB
        LI Ø,ARG
        BL @FFAC
        MOV @D255,@FAC
        BLWP @XMLLNK
        DATA CIF
        BLWP @XMLLNK
        DATA FDIV
        LI Ø,DX
        BL @FFAC
        LI Ø,YO             DY=(YO-YU)/191 Y-ABSTAND ZWISCHEN ZWEI PUNKTEN
        BL @TARG                           (BEZUEGLICH DES BILDSCHIRMS)
        LI Ø,YU
        BL @TFAC
        BLWP @XMLLNK
        DATA FSUB
        LI Ø,ARG
        BL @FFAC
        MOV @D191,@FAC
        BLWP @XMLLNK
        DATA CIF
        BLWP @XMLLNK
        DATA FDIV
        LI Ø,DY
        BL @FFAC
        MOV @D2,@FAC        2 -> REAL2
        BLWP @XMLLNK
        DATA CIF
        LI Ø,REAL2
        BL @FFAC
        CLR 9               R9=Ø  Y-WERT ZAEHLER
START1  CLR 1Ø              R1Ø=Ø X-WERT ZAEHLER
START2  LI Ø,DX             XC=XU+R1Ø*DX AKTUELLER X-WERT IM KORDINATENSYSTEM
```

```
      BL §TARG
      MOV 1Ø,§FAC
      BLWP §XMLLNK
      DATA CIF
      BLWP §XMLLNK
      DATA FMUL
      LI Ø,XU
      BL §TARG
      BLWP §XMLLNK
      DATA FADD
      LI Ø,XC
      BL §FFAC
      LI Ø,DY          YC=YU+R9*DY AKTUELLER Y-WERT IM KOORDINATENSYSTEM
      BL §TARG
      MOV 9,§FAC
      BLWP §XMLLNK
      DATA CIF
      BLWP §XMLLNK
      DATA FMUL
      LI Ø,YU
      BL §TARG
      BLWP §XMLLNK
      DATA FADD
      LI Ø,YC
      BL §FFAC
      CLR 8            R8=Ø ITERATIONSZAEHLER
      CLR §XZ          XZ=Ø INITIALISIEREN DER KOMPLEXEN ZAHL MIT DEN KOMPO-
      CLR §YZ          YZ=Ø NENTEN XZ UND YZ (RESTLICHE 6 BYTES UNWICHTIG)
START3 INC 8           ERHOEHEN DES ITERATIONSZAEHLERS
      LI Ø,XZ          XX=XZ*XZ QUADRAT DER X-KOMPONENTE
      BL §TARG
      LI Ø,XZ
      BL §TFAC
      BLWP §XMLLNK
      DATA FMUL
      LI Ø,XX
      BL §FFAC
      LI Ø,YZ          YY=YZ*YZ QUADRAT DER Y-KOMPONENTE
      BL §TARG
      LI Ø,YZ
      BL §TFAC
      BLWP §XMLLNK
      DATA FMUL
      LI Ø,YY
      BL §FFAC
      LI Ø,XZ          YZ=2*XZ*YZ-YC NEUER IMAGINAERTEIL  -I
      BL §TARG                                              I
      LI Ø,REAL2                                            I
      BL §TFAC                                              I
      BLWP §XMLLNK                                          I
      DATA FMUL                                             I
      LI Ø,YZ                                               I
      BL §TARG                                              I
      BLWP §XMLLNK                                          I   DER KOMPLE-
      DATA FMUL                                              I- XEN ZAHL
      LI Ø,ARG                                              I
      BL §FFAC                                              I
      LI Ø,YC                                               I
```

```
        BL §TFAC                                                    I
        BLWP §XMLLNK                                                I
        DATA FSUB                                                   I
        LI Ø,YZ                                                     I
        BL §FFAC                                                    I
        LI Ø,YY         XZ=XX-(YY+XC) NEUER REALTEIL               -I
        BL §TARG
        LI Ø,XC
        BL §TFAC
        BLWP §XMLLNK
        DATA FADD
        LI Ø,XX
        BL §TARG
        BLWP §XMLLNK
        DATA FSUB
        LI Ø,XZ
        BL §FFAC
        C 7,8           MAXIMALE ANZAHL VON ITERATIONEN DURCHGEFUEHRT ?
        JEQ START4      WENN JA NAECHSTER PUNKT
        LI Ø,XX         ABSOLUTWERT DER KOMPLEXEN ZAHL (^2) ERRECHNEN: XX+YY
        BL §TARG
        LI Ø,YY
        BL §TFAC
        BLWP §XMLLNK
        DATA FADD
        LI Ø,R1ØØ       KLEINER ALS 1Ø (^2)
        BL §TARG
        BLWP §XMLLNK
        DATA FCOM
        MOVB §>837C,Ø
        SLA Ø,2
        JOC START6      WENN JA NAECHSTE ITERATION
        MOV 1Ø,Ø        BILDSCHIRM-X-POSITION
        MOV 9,1         BILDSCHIRM-Y-POSITION
        MOV 8,2         FARBE (UNGERADE-WEISS/GERADE-SCHWARZ)
        INC 2           INVERTIERTES BILDES WENN WEGGELASSEN
        BLWP §PLOT      PUNKT DARSTELLEN
START4  INC 1Ø          NAECHSTE SPALTE DES BILDSCHIRMS
        CI 1Ø,256       ENDE DER ZEILE ERREICHT ?
        JEQ $+6         WENN JA NAECHSTE ZEILE
        B §START2       SONST NAECHSTEN PUNKT
        INC 9           NAECHSTE ZEILE
        CI 9,192        ENDE DES BILDSCHIRMS ERREICHT ?
        JEQ START5      WENN JA AUFHOEHREN
        B §START1       SONST WEITERMACHEN
START5  CLR Ø           TASTATURABFRAGE
        MOVB Ø,§>8374
        BLWP §KSCAN
        MOVB §>8375,Ø
        CI Ø,>ØDØØ      ENTER-TASTE ?
        JNE START5      WENN NEIN ZURUECK
        BLWP §SAVE      SONST ABSPEICHERN (ENTFAELLT,WENN UNERWUENSCHT)
        BLWP Ø          SOFTWARE-RESET
START6  B §START3       RUECKSPRUNG FUER JOC, DA ENTFERNUNG ZU GROSS

        END
```

```
*  *********************************
* ***********************************
* *          2.TEIL: DSKSAVE        *
* *               BY                *
* *        KARSTEN KUTOWSKI         *
* * ASSEMBLIERT: 'DSKSAVEO'         *
* ***********************************
*  *********************************

       AORG >7F00

VMBW   EQU >6028
DSRLNK EQU >6038

       DEF SAVE

PAB    DATA >0000
       DATA >0100
       DATA >060A
       DATA >0000
       DATA >4040
       DATA >1800
       DATA >000F
       TEXT 'DSK1.MANDELBROT'
       EVEN

SAVE   DATA WORK,SAVE1

WORK   BSS 32

SAVE1  LI 0,>3400
       LI 1,PAB
       LI 2,29
       BLWP §VMBW
       LI 0,>340D
       MOV 0,§>8356
       BLWP §DSRLNK
       DATA 8
       RTWP

       END
```

```
*  *********************************
* ***********************************
* *          3.TEIL: DSKLOAD        *
* *               BY                *
* *        KARSTEN KUTOWSKI         *
* * ASSEMBLIERT: 'DSKLOADO'         *
* ***********************************
*  *********************************

       AORG >7E00

VSBW   EQU >6024
VMBW   EQU >6028
VWTR   EQU >6034
KSCAN  EQU >6020
STRREF EQU >604C
DSRLNK EQU >6038

       DEF LOAD

PAB    DATA >0000
       DATA >0100
       DATA >050A
       DATA >0000
       DATA >4040
       DATA >1800
       DATA >000B
PAB1   BSS 16

RDATA  DATA >0002,>01E0,>0206,>03FF
       DATA >0403,>0536,>0607,>07F1

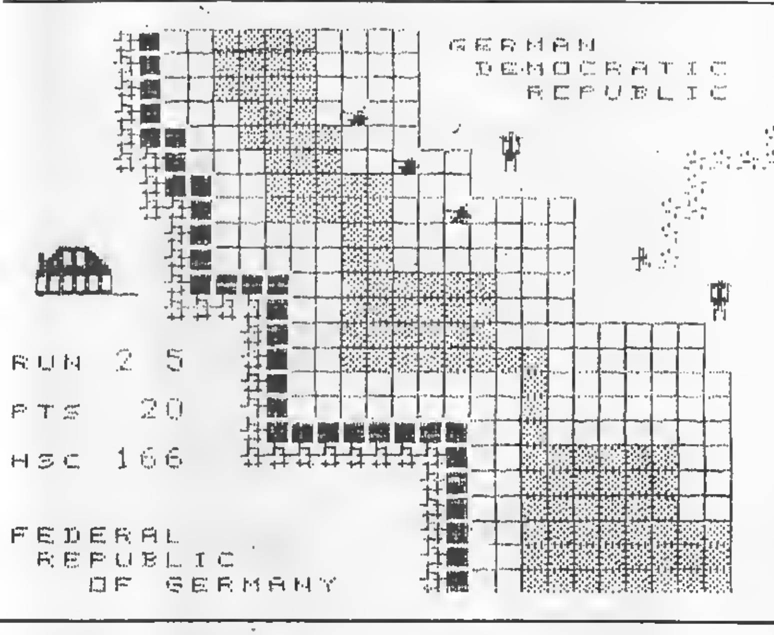
BITMAP DATA WS,BITMA1

WS     BSS 32

BITMA1 LI 1,RDATA
       MOVB §3(1),§>83D4
BITMA2 MOV *1+,0
       BLWP §VWTR
       CI 1,RDATA+16
       JNE BITMA2
       CLR 0
       CLR 1
BITMA3 BLWP §VSBW
       INC 0
       CI 0,>1800
       JNE BITMA3
BITMA4 BLWP §VSBW
       AI 1,>0100
       INC 0
       CI 0,>1B00
       JNE BITMA4
       LI 1,>D000
       BLWP §VSBW
       LI 0,>2000
       LI 1,>F100
BITMA5 BLWP §VSBW
       INC 0
       CI 0,>3800
       JNE BITMA5
       RTWP
LOAD   CLR 0
       LI 1,1
       LI 2,PAB1-1
       LI 3,>0F00
       MOVB 3,*2
       BLWP §STRREF
       BLWP §BITMAP
       LI 0,>1C00
       LI 1,PAB
       LI 2,25
       BLWP §VMBW
       LI 0,>1C0D
       MOV 0,§>8356
       BLWP §DSRLNK
       DATA 8
LOOP2  CLR 0
       MOVB 0,§>8374
       BLWP §KSCAN
       MOVB §>8375,0
       CI 0,>0D00
       JNE LOOP2
       BLWP §0
       END
```

WESTLICH DES STACHEL-DRAHTS

Westlich des Stacheldrahts, oder die Simulation einer Flucht über die deutsch/deutsche Grenze, ist ein Programm, das sich durch seine wirklichkeitsnahe Aktualität sehr stark von den Schieß- und-Stirb-Spielen abhebt.
Westlich des Stacheldrahts ist keine Verherrlichung von Gewalt und erst recht nicht von oben genannten Szenen.



Westlich des Stacheldrahts ist ein kritisches Programm, das den User anregen soll, sich der Situation klar zu werden, in der er sich während des Spiels befindet, und sich zu verdeutlichen, daß dies kein Science-fiction-Märchen, sondern bittere Realität ist.
Die Hauptpersonen in „Westlich des Stacheldrahts" ist ein DDR-Bürger, der lediglich versucht, seinen Traum vom goldenen Westen in die Realität umzusetzen.
Wäre da nicht der Stacheldraht . . .
Dargestellt ist ein Grenzabschnitt mit den 5 Grenzstreifen: Stacheldraht, Todesstreifen, Sichtstreifen, Minensperre, Sicherungsstreifen.
Ziel des Spiels ist es, diese zu überwinden. Hört sich zwar einfach an, aber die Erfolgsquote ist kaum höher als in der Realität.
Gewürfelt wird mit >FIRE< bzw. mit >ENTER<. Dann wird der RUNNER bewegt.
Sie müssen versuchen, eine Öffnung im Zaun zu finden, die bei jedem Rückschlag neu definiert wird. Dazu gilt es zunächst, den SICHERUNGSSTREIFEN zu überqueren, was zu Anfang noch ziemlich einfach ist. Danach betreten Sie die MINENSPERRE (max. 12 Minen) und werden nach Betreten einer Mine zurückgeschlagen. Haben Sie dieses Feld jedoch schadlos überstanden, gelangen Sie auf den SICHTSTREIFEN. Danger. Ist Ihr RUNWERT gleich dem WURFWERT, werden Sie um 7 Felder zurückgeworfen. Was Sie da erwartet... (Keine Neudefinition der Öffnung!)
Danach folgt der TODESSTREIFEN, der identisch mit der MINENSPERRE ist, jedoch eine größere Anzahl von Minen besitzt. Versuchen Sie nun die Öffnung im STACHELDRAHT zu finden, was mit einer SOUNDORTUNG(!) möglich ist. Mit der zunehmenden Tiefe des Tones nähern Sie sich der Öffnung. ATTENTION. Sie befinden sich auf dem TODESSTREIFEN...
Haben Sie die Öffnung gefunden, sind Sie nicht unbedingt gerettet. Vielleicht war es eine Falle und eine TRETMINE explodiert! Andernfalls begeben Sie sich so schnell wie möglich auf das Haus zu. Sie sind gerettet wenn der RUNWERT gleich dem WURFWERT ist.
Zu erwähnen wäre noch: Werden Sie zurückgeschlagen, erscheint ein PANZER und der Schwierigkeitsgrad des 'Spiels' nimmt zu. Stehen Sie jetzt in einer Linie mit einem PANZER,bei identischem WURF- und RUNWERT, eröffnet dieser das Feuer . . . !
Ist die Anzahl der PANZER auf 8 angestiegen, sterben Sie bei erneutem Schußwechsel, und Ihre POINTS gehen nicht in die HIGHSCOREWERTUNG ein, die hier logischerweise(!) die niedrigste Punktzahl wertet.
— Versuchen Sie nicht, von WESTEN gegen den STACHELDRAHT zu laufen.
Das könnte Ihre Träume zerschlagen!

AUCH OHNE JOYSTICK RICHTUNG WESTEN...

„Westlich des Stacheldrahts" wird mit JOYSTICK 2 gesteuert.
Diejenigen, die ohne Joystick in den Westen wollen, müssen folgende Zeilen umändern:

```
1290  CALL KEY(0.K.S)
1310  IF K=13 THEN 1320 ELSE 1290
1540  CALL KEY(0.K.S)
1550  IF S=0  THEN 1540
1560  IF K=69 THEN 1570 ELSE 1630
1630  IF K=88 THEN 1640 ELSE 1700
1700  IF K=83 THEN 1710 ELSE 1770
1770  IF K=68 THEN 1780 ELSE 1840
```

FUNKTIONSBLÖCKE

00100–00120 TITEL
00150–00340 ASCII DEF
00345–00560 COLOR
00570–00970 BILDAUFBAU
00980–01140 INTRO
01150–01270 VORBEREITUNG
01280–01400 WURF
01410–01460 DISPLAY AT
01470–02200 RUN/POS
02210–03050 SEKTORENFESTLEGUNG
03060–03150 SOUNDORTUNG
03160–04240 POSITIONIERUNG
04250–04280 TRETMINE
04290–04620 MINENVERTEILUNG
04630–05090 PANZERROUTINE
05100–10000 NACHTBILD

```
10 REM*******************
11 REM*Westl.des Stachel-*
12 REM*    Drahtes       *
13 REM*  Copyright  by   *
14 REM* Thomas Jerusalem *
15 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM* optional:XBasic  *
20 REM*                  *
21 REM*                  *
23 REM* Speicherbelegung *
24 REM*   12036 Bytes    *
25 REM*******************
26 REM
110 CALL CLEAR
120 PRINT "WESTLICH DES STAC
HELDRAHTS": : :".........c_
DateControl 1984"
150 RANDOMIZE
170 REM
180 CALL CHAR(34,"000A58783C
7F380C")
190 CALL CHAR(128,"FFFF99999
999FFFF")
200 CALL CHAR(36,"0000000000
0000FF")
210 CALL CHAR(37,"0103676F7F
7F7FFF")
220 CALL CHAR(38,"80C0E0F0F8
FCFEFF")
230 CALL CHAR(89,"0000020CFE
1F7F3E")
240 CALL CHAR(33,"0828200318
026200")
250 CALL CHAR(136,"000000003
03C1F3F")
260 CALL CHAR(137,"7F7E3E7CF
8F0E0C0")
270 CALL CHAR(40,"203C24EF04
247E24")
280 CALL CHAR(64,"00FEFEFEFE
FEFEFE")
290 CALL CHAR(72,"FF01010101
010101")
300 CALL CHAR(80,"FF01010101
010101")
310 CALL CHAR(41,"0000187EFF
7E5A5A")
320 CALL CHAR(42,"7E7E3C3C3C
242424")
330 CALL CHAR(88,"FF01010101
010101")
340 CALL CHAR(43,"18181C7C18
1E1230")
345 CALL CLEAR
350 CALL SCREEN(4)
360 GOTO 480
370 CALL COLOR(1,2,1)
380 FOR I=2 TO 5
390 CALL COLOR(I,2,1)
400 NEXT I
410 CALL COLOR(6,2,11)
420 CALL COLOR(7,2,13)
430 CALL COLOR(8,2,12)
440 FOR I=9 TO 13
450 CALL COLOR(I,2,1)
460 NEXT I
470 RETURN
480 REM
490 FOR I=1 TO 13
500 CALL COLOR(I,4,4)
510 NEXT I
520 REM
530 GOSUB 3920
540 REM
550 GOSUB 4290
560 GOSUB 4450
570 REM
580 PRINT "....(@HHPPPPXXXX
german.........(@HHPPPPXXXX
 democratic....(@HHPPPPXXXX.
...republic"
590 PRINT "....(@HHHPPPXXXX"
:"....(@@HHPPPPXXX":"....((@
HHHPPPXXXXX":".....(@@HHPPPP
PXXX"
600 PRINT ".....((@HHPPPPPXX
XXXXX":"......(@HHHHHPPXXXXX
XX":"......(@HHHHHPPXXXXXXX"
610 PRINT "......(@@@@HHPPPP
PPXXX":"......((((@HHPPPPPPX
XX":".........(@HHPPPPPPXXXX
XXXX"
620 PRINT ".........(@HHPPPP
PPPPXXXXXX"
630 FOR I=1 TO 2
640 PRINT ".........(@HHHHHH
HHHPXXXXXXX"
650 NEXT I
660 PRINT ".........(@@@@@@@
@HHPXXXXXXX.........((((((((
@HHPPPPPPXX................(
@HHPPPPPPXX"
670 PRINT "................(
@HHPPPPPPXX"
680 PRINT "federal.........(
@HHPPPPPPPP republic.......(
@HHPPPPPPPP...of germany...(
@HHPPPPPPPP"
690 CALL HCHAR(DOORZ,DOORS,4
0)
700 CALL HCHAR(11,4,128,3)
710 CALL HCHAR(11,7,36)
720 CALL HCHAR(10,4,37)
730 CALL HCHAR(10,5,128)
740 CALL HCHAR(10,6,38)
750 CALL HCHAR(6,22,42)
760 CALL HCHAR(5,22,41)
770 CALL HCHAR(12,30,42)
```

LISTINGS

ANZEIGENSERVICE

Die große Börse für jeden Zweck in der TI REVUE. Kostenlos für Privat-Inserenten. Sportbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso der Preis für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht TI-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht angenommen. Für Privatanbieter: maximal acht Zeilen à 28 Anschläge. Für gewerbliche Anbieter: 5 DM p. mm.

TI REVUE
Anzeigenabteilung
Postfach 1107
8044 Lohhof

Name ______________________

Vorname ______________________

Straße/Hausnr. ______________________

PLZ/Ort ______________________

ABO-SERVICE-KARTE

COUPON

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten zwölf Ausgaben an untenstehende Anschrift. Sollte ich nicht vier Wochen vor Ablauf schriftlich kündigen, läuft diese Abmachung automatisch weiter.

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

Name ______________________

Vorname ______________________

Straße/Hausnr. ______________________

PLZ/Ort ______________________

Ich bezahle:

☐ per beiliegendem Verrechnungsscheck

☐ gegen Rechnung

☐ bargeldlos per Bankeinzug von meinem Konto (TI 9/86)

bei (Bank) und Ort ______________________

Kontonummer ______________________

Bankleitzahl ______________________
(steht auf jedem Kontoauszug)

Unterschrift ______________________

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift

TI REVUE
Abo-Service 9/86
Postfach 1107
8044 UNTERSCHLEISSHEIM

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen
die Listings dieses Heftes auf
☐ Diskette zum Preis von (25,– DM)

Ich zahle:
Bar – per beigefügtem Geld ( )
per beigefügtem Scheck ( )
Gegen Bankabbuchung am Versandtag ( )
Zutreffendes bitte ankreuzen!

Meine Bank (mit Ortsname) ....

Meine Kontonummer ....

Meine Bankleitzahl .... (steht auf jedem Bankauszug)

Vorname .... Nachname ....

Str./Nr. .... PLZ / Ort 9/86

Hiermit bestätige ich mit meiner Unterschrift, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren.

Unterschrift ....

TI-REVUE
KASSETTENSERVICE 9/86
Postfach 1107
8044 Unterschleißheim

Verkaufsbedingungen: Versand nur gegen Vorauskasse oder Bankabbuchung. Umtauschrecht bei Nichtfunktionieren. Keine Nachnahme.

RESERVIERUNGS-SERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen

( ) Exemplar(e) TI SPECIAL (Nr. 4) 14,80 ( ) Exemplar(e) TI ASSEMBLER SPECIAL 19,80

Zutreffendes bitte ankreuzen!
Ich zahle:
per beigefügtem Scheck / Schein ( )
Gegen Bankabbuchung am Versandtag ( )
Meine Bank (mit Ortsname) 9/86

Meine Kontonummer ....

Meine Bankleitzahl .... (steht auf jedem Bankauszug)

Vorname .... Nachname ....

Str./Nr. .... PLZ / Ort ....

Verkaufsbedingungen: Versand nur gegen Vorkasse oder Bankabbuchung. Keine Nachnahme!

Unterschrift ....

Bitte ausschneiden und einsenden an
TI-REVUE
Special-Service 9/86
Postfach 1107
8044 Unterschleißheim

VERDIENEN SIE GELD MIT IHREM COMPUTER!

Haben **Haben Sie einen** TI 99/A? Können Sie programmieren? In Basic oder Maschinensprache? Dann bietet TI-REVUE Ihnen die Möglichkeit, damit Geld zu verdienen.

Wie? Ganz einfach. Sie senden uns die Programme, die Sie für einen Abdruck als geeignet halten, zusammen mit einer Kurzbeschreibung, aus der auch die verwendete Hardware – eventuelle Erweiterungen – benutzte Peripherie – hervorgehen muß (Schauen Sie sich dazu den Kopf unserer Programmlistings an.)

Benötigt werden: Zwei Listings des Programms sowie eine Datenkassette oder Diskette! Wenn die Redaktion sich überzeugt hat, daß dieses Programm läuft und sich zum Abdruck eignet, zahlen wir Ihnen pro Programm je nach Umfang bis zu DM 300,–! Sollten Sie keinen Drucker haben, genügt der Datenträger.

Sie erhalten Ihre Kassette/Diskette selbstverständlich zurück, wenn Sie einen ausreichend frankierten Rückumschlag mit Ihrer Adresse beifügen.

Bei der Einsendung müssen Sie mit Ihrer Unterschrift garantieren, daß Sie der alleinige Inhaber der Urheber-Rechte sind! Benutzen Sie bitte anhängendes Formular! (Wir weisen darauf hin, daß auch die Redaktion amerikanische und englische Fachzeitschriften liest und „umgestaltete" Programme ziemlich schnell erkennt).

Um Ihnen die Arbeit zu erleichtern, finden Sie hier ein Formular. Sie können es ausschneiden oder fotokopieren.

PROGRAMMANGEBOT

Name des Einsenders: ____________________

Straße/Hausnr./Tel.: ____________________

Plz/Ort: ____________________

Hiermit biete ich Ihnen zum Abdruck folgende(s) Programm(e) an:

____

____

____

____

Benötigte Geräte: ____________________

____

Beigefügt ( ) Listings ( ) Kassette ( ) Diskette

Ich versichere, der alleinige Urheber des Programmes zu sein!
Hiermit ermächtige ich die Redaktion, dieses Programm abzudrucken und wirtschaftlich zu verwerten. Sollte es in den Kassetten-Service aufgenommen werden, erhalte ich auch dafür eine entsprechende Vergütung, das Copyright geht auf den Verlag über.

____

Rechtsverbindliche Unterschrift

TI-REVUE
Postfach 1107
8044 Lohhof

```
78Ø CALL HCHAR(11,3Ø,41)
79Ø B$="run"
8ØØ ZZ=14
81Ø SS=2
82Ø GOSUB 141Ø
83Ø B$="pts"
84Ø ZZ=16
85Ø SS=2
86Ø GOSUB 141Ø
87Ø B$="hsc"
88Ø ZZ=18
89Ø SS=2
9ØØ GOSUB 141Ø
91Ø B$=STR$(HS)
92Ø ZZ=18
93Ø SS=6
94Ø GOSUB 141Ø
95Ø GOSUB 37Ø
96Ø FOR D=1 TO 1ØØØ
97Ø NEXT D
98Ø REM
99Ø CALL HCHAR(5,32,43)
1ØØØ CALL HCHAR(5,32,33)
1Ø1Ø FOR I=32 TO 29 STEP -1
1Ø2Ø CALL HCHAR(6,I,43)
1Ø3Ø CALL HCHAR(6,I,33)
1Ø4Ø NEXT I
1Ø5Ø CALL HCHAR(7,29,43)
1Ø6Ø CALL HCHAR(7,29,33)
1Ø7Ø CALL HCHAR(8,29,43)
1Ø8Ø CALL HCHAR(8,29,33)
1Ø9Ø FOR I=8 TO 1Ø
11ØØ CALL HCHAR(I,28,43)
111Ø CALL HCHAR(I,28,33)
112Ø NEXT I
113Ø CALL HCHAR(1Ø,27,43)
114Ø CALL HCHAR(1Ø,27,33)
115Ø REM
116Ø RZ=1Ø
117Ø RS=26
118Ø CALL HCHAR(RZ,RS,43)
119Ø GOSUB 128Ø
12ØØ GOTO 147Ø
121Ø RX=RX+1
122Ø ZAE=ZAE+1
123Ø GOSUB 192Ø
124Ø PTS=PTS+1
125Ø GOSUB 215Ø
126Ø GOTO 147Ø
127Ø GOTO 127Ø
128Ø REM
129Ø CALL KEY(2,K,S)
13ØØ IF S=Ø THEN 129Ø
131Ø IF K=18 THEN 132Ø ELSE
129Ø
132Ø REM
133Ø WURF=INT(RND*6)+1
134Ø ZAE=Ø
135Ø RX=Ø
136Ø B$=STR$(WURF)
137Ø ZZ=14
138Ø SS=6
139Ø GOSUB 141Ø
14ØØ RETURN
141Ø REM DISPLAY
142Ø A$=B$
143Ø FOR XX=1 TO LEN(A$)
144Ø CALL HCHAR(ZZ,SS+XX,ASC
(SEG$(A$,XX,1)))
145Ø NEXT XX
146Ø RETURN
147Ø REM RUN
148Ø IF (RZ=11)*(RS=7)THEN 1
49Ø ELSE 152Ø
149Ø CALL HCHAR(11,7,36)
15ØØ RS=8
151Ø CALL HCHAR(RZ,RS,43)
152Ø IF PTS=1ØØ THEN 153Ø EL
SE 154Ø
153Ø GOSUB 51ØØ
154Ø CALL JOYST(2,X,Y)
156Ø IF Y=4 THEN 157Ø ELSE 1
63Ø
157Ø CALL HCHAR(RZ,RS,33)
158Ø RZ=RZ-1
159Ø GOSUB 221Ø
16ØØ GOSUB 236Ø
161Ø CALL HCHAR(RZ,RS,43)
162Ø GOTO 185Ø
163Ø IF Y=-4 THEN 164Ø ELSE
17ØØ
164Ø CALL HCHAR(RZ,RS,33)
165Ø RZ=RZ+1
166Ø GOSUB 221Ø
167Ø GOSUB 236Ø
168Ø CALL HCHAR(RZ,RS,43)
169Ø GOTO 185Ø
17ØØ IF X=-4 THEN 171Ø ELSE
177Ø
171Ø CALL HCHAR(RZ,RS,33)
172Ø RS=RS-1
173Ø GOSUB 221Ø
174Ø GOSUB 236Ø
175Ø CALL HCHAR(RZ,RS,43)
176Ø GOTO 185Ø
177Ø IF X=4 THEN 178Ø ELSE 1
84Ø
178Ø CALL HCHAR(RZ,RS,33)
179Ø RS=RS+1
18ØØ GOSUB 221Ø
181Ø GOSUB 236Ø
182Ø CALL HCHAR(RZ,RS,43)
183Ø GOTO 185Ø
184Ø GOTO 154Ø
185Ø IF RX+1<WURF THEN 121Ø
186Ø GOSUB 475Ø
187Ø ZAE=ZAE+1
188Ø GOSUB 192Ø
189Ø PTS=PTS+1
19ØØ GOSUB 215Ø
```

```
1910 GOTO 1980
1920 REM ZAE
1930 B$=STR$(ZAE)
1940 ZZ=14
1950 SS=8
1960 GOSUB 1410
1970 RETURN
1980 IF (RZ=11)*(RS=7)THEN 1
990 ELSE 2070
1990 CALL COLOR(13,2,11)
2000 CALL HCHAR(11,7,36)
2010 GOSUB 2090
2020 B$="saved"
2030 ZZ=14
2040 SS=6
2050 GOSUB 1410
2060 GOTO 4940
2070 GOSUB 2090
2080 GOTO 2140
2090 B$="..."
2100 ZZ=14
2110 SS=6
2120 GOSUB 1410
2130 RETURN
2140 GOTO 1190
2150 REM
2160 B$=STR$(PTS)
2170 ZZ=16
2180 SS=6
2190 GOSUB 1410
2200 RETURN
2210 REM MAX.
2220 IF RZ=0 THEN 2230 ELSE
2250
2230 RZ=RZ+1
2240 GOTO 2350
2250 IF RZ=24 THEN 2260 ELSE
 2280
2260 RZ=RZ-1
2270 GOTO 2350
2280 IF RS=33 THEN 2290 ELSE
 2310
2290 RS=RS-1
2300 GOTO 2350
2310 IF RS=0 THEN 2320 ELSE
2340
2320 RS=RS+1
2330 GOTO 2350
2340 GOTO 2350
2350 RETURN
2360 REM
2370 IF (RZ>0)*(RZ<13)*(RS>6
)*(RS<11)THEN 2430
2380 IF (RZ>0)*(RZ<19)*(RS>1
0)*(RS<16)THEN 2540
2390 IF (RZ>0)*(RZ<24)*(RS>1
5)*(RS<21)THEN 2710
2400 IF (RZ>7)*(RZ<24)*(RS>2
0)*(RS<26)THEN 2830
2410 IF (RZ>12)*(RZ<24)*(RS>
25)*(RS<31)THEN 2940
2420 RETURN
2430 REM
2440 IF (RZ>0)*(RZ<7)*(RS=7)
THEN 3010
2450 IF (RZ>5)*(RZ<9)*(RS=8)
THEN 3010
2460 IF (RZ>7)*(RZ<13)*(RS=9
)THEN 3010
2470 IF (RZ=12)*(RS=10)THEN
3160
2480 IF (RZ>0)*(RZ<6)*(RS=8)
THEN 3360
2490 IF (RZ>4)*(RZ<8)*(RS=9)
THEN 3360
2500 IF (RZ>6)*(RZ<12)*(RS=1
0)THEN 3360
2510 IF (RZ>0)*(RZ<5)*(RS=9)
THEN 3720
2520 IF (RZ>0)*(RZ<7)*(RS=10
)THEN 3720
2530 RETURN
2540 REM
2550 IF (RZ=12)*(RS=11)THEN
3160
2560 IF (RZ>11)*(RZ<19)*(RS=
12)THEN 3010
2570 IF (RZ=18)*(RS>12)*(RS<
16)THEN 3160
2580 IF (RZ=11)*(RS>10)*(RS<
14)THEN 3360
2590 IF (RZ>11)*(RZ<18)*(RS=
13)THEN 3360
2600 IF (RZ=17)*(RS>13)*(RS<
16)THEN 3360
2610 IF (RZ>3)*(RZ<11)*(RS=1
1)THEN 3720
2620 IF (RZ>5)*(RZ<11)*(RS=1
2)THEN 3720
2630 IF (RZ>8)*(RZ<11)*(RS=1
3)THEN 3720
2640 IF (RZ>8)*(RZ<17)*(RS>1
3)*(RS<16)THEN 3720
2650 IF (RZ>0)*(RZ<4)*(RS>10
)*(RS<15)THEN 3770
2660 IF (RZ>3)*(RZ<6)*(RS>11
)*(RS<15)THEN 3770
2670 IF (RZ=5)*(RS=15)THEN 3
770
2680 IF (RZ>5)*(RZ<9)*(RS>12
)*(RS<16)THEN 3770
2690 IF (RZ>0)*(RZ<5)*(RS=15
)THEN 3900
2700 RETURN
2710 REM
2720 IF (RZ=18)*(RS>15)*(RS<
20)THEN 3160
2730 IF (RZ>18)*(RZ<24)*(RS=
19)THEN 3010
2740 IF (RZ=17)*(RS>15)*(RS<
```

```
21)THEN 3360
2750 IF (RZ>17)*(RZ<24)*(RS=
20)THEN 3360
2760 IF (RZ>14)*(RZ<17)*(RS>
15)*(RS<21)THEN 3720
2770 IF (RZ>6)*(RZ<15)*(RS>1
5)*(RS<18)THEN 3770
2780 IF (RZ>10)*(RZ<15)*(RS>
17)*(RS<21)THEN 3770
2790 IF (RZ>0)*(RZ<7)*(RS>15
)*(RS<19)THEN 3900
2800 IF (RZ=6)*(RS>18)*(RS<2
1)THEN 3900
2810 IF (RZ>6)*(RZ<11)*(RS>1
7)*(RS<21)THEN 3900
2820 RETURN
2830 REM
2840 IF (RZ>14)*(RZ<24)*(RS>
20)*(RS<23)THEN 3720
2850 IF (RZ>10)*(RZ<15)*(RS=
21)THEN 3770
2860 IF (RZ=14)*(RS=22)THEN
3770
2870 IF (RZ>13)*(RZ<24)*(RS=
23)THEN 3770
2880 IF (RZ>17)*(RZ<24)*(RS>
23)*(RS<26)THEN 3770
2890 IF (RZ>7)*(RZ<11)*(RS=2
1)THEN 3900
2900 IF (RZ>7)*(RZ<14)*(RS>2
1)*(RS<25)THEN 3900
2910 IF (RZ=13)*(RS=25)THEN
3900
2920 IF (RZ>13)*(RZ<18)*(RS>
23)*(RS<26)THEN 3900
2930 RETURN
2940 REM
2950 IF (RZ>17)*(RZ<24)*(RS>
25)*(RS<29)THEN 3770
2960 IF (RZ>20)*(RZ<24)*(RS>
28)*(RS<31)THEN 3770
2970 IF (RZ>12)*(RZ<18)*(RS>
25)*(RS<30)THEN 3900
2980 IF (RZ>14)*(RZ<21)*(RS=
30)THEN 3900
2990 IF (RZ>17)*(RZ<21)*(RS=
29)THEN 3900
3000 RETURN
3010 REM
3020 GOSUB 3060
3030 IF (RZ=DOORZ)*(RS=DOORS
)THEN 3210
3040 RS=RS+1
3050 GOTO 3350
3060 REM
3070 SOU=23-(23-DOORZ)
3080 IF RZ>DOORZ THEN 3120
3090 IF RZ<DOORZ THEN 3140 E
LSE 3100
3100 CALL SOUND(-100,-1,0)
3110 GOTO 3150
3120 CALL SOUND(-100,(RZ-SOU
)*110,0)
3130 GOTO 3150
3140 CALL SOUND(-100,(SOU-RZ
)*110,0)
3150 RETURN
3160 REM
3170 GOSUB 3060
3180 IF (RZ=DOORZ)*(RS=DOORS
)THEN 3210
3190 RZ=RZ-1
3200 GOTO 3350
3210 IF MINED=1 THEN 3220 EL
SE 3310
3220 FOR I=0 TO 30 STEP 3
3230 CALL SOUND(100,-5,I)
3240 NEXT I
3245 CALL HCHAR(RZ,RS,34)
3280 GOSUB 4630
3290 RZ=10
3300 RS=26
3310 REM
3311 GOSUB 5300
3320 GOSUB 3920
3330 REM
3340 MINED=INT(RND*2)+1
3350 RETURN
3360 GOSUB 5260
3370 IF (RZ<10)+(RZ=11)*(RS=
10)THEN 3500
3380 IF (RZ<17)+(RZ=17)*(RS>
12)*(RZ=17)*(RS<16)THEN 3530
3390 GOTO 3560
3400 IF (RZ=MI1Z(1))*(RS=MI1
S(1))+(RZ=MI1Z(2))*(RS=MI1S(
2))+(RZ=MI1Z(3))*(RS=MI1S(3)
)THEN 3520
3410 IF (RZ=MI1Z(4))*(RS=MI1
S(4))+(RZ=MI1Z(5))*(RS=MI1S(
5))THEN 3520
3420 RETURN
3430 IF (RZ=MI2Z(1))*(RS=MI2
S(1))+(RZ=MI2Z(2))*(RS=MI2S(
2))+(RZ=MI2Z(3))*(RS=MI2S(3)
)THEN 3550
3440 IF (RZ=MI2Z(4))*(RS=MI2
S(4))+(RZ=MI2Z(5))*(RS=MI2S(
5))THEN 3550
3450 RETURN
3460 IF (RZ=MI3Z(1))*(RS=MI3
S(1))+(RZ=MI3Z(2))*(RS=MI3S(
2))+(RZ=MI3Z(3))*(RS=MI3S(3)
)THEN 3580
3470 IF (RZ=MI3Z(4))*(RS=MI3
S(4))+(RZ=MI3Z(5))*(RS=MI3S(
5))THEN 3580
3480 RETURN
3490 GOTO 3710
3500 GOSUB 3400
```

```
351Ø RETURN
352Ø GOTO 359Ø
353Ø GOSUB 343Ø
354Ø RETURN
355Ø GOTO 359Ø
356Ø GOSUB 346Ø
357Ø RETURN
358Ø GOTO 359Ø
359Ø REM EXPL
36ØØ FOR I=Ø TO 3Ø STEP 5
361Ø CALL SOUND(-1ØØ,-5,I)
362Ø CALL SOUND(-1ØØ,-6,I)
363Ø NEXT I
364Ø CALL HCHAR(RZ,RS,34)
365Ø GOSUB 463Ø
366Ø RZ=1Ø
367Ø RS=26
368Ø CALL HCHAR(RZ,RS,43)
369Ø REM
37ØØ GOSUB 392Ø
371Ø RETURN
372Ø GOSUB 526Ø
373Ø IF ZAE=WURF-1 THEN 375Ø
374Ø RETURN
375Ø RS=RS+7
376Ø RETURN
377Ø GOSUB 526Ø
378Ø IF (RZ<9)*(RS<15)THEN 3
82Ø
379Ø IF (RZ<15)*(RS<21)THEN
384Ø
38ØØ IF (RS<26)THEN 386Ø
381Ø GOTO 388Ø
382Ø IF (RZ=TP1Z(1))*(RS=TP1
S(1))+(RZ=TP1Z(2))*(RS=TP1S(
2))+(RZ=TP1Z(3))*(RS=TP1S(3)
)THEN 359Ø
383Ø RETURN
384Ø IF (RZ=TP2Z(1))*(RS=TP2
S(1))+(RZ=TP1Z(2))*(RS=TP1S(
2))+(RZ=TP1Z(3))*(RS=TP1S(3)
)THEN 359Ø
385Ø RETURN
386Ø IF (RZ=TP3Z(1))*(RS=TP3
S(1))+(RZ=TP3Z(2))*(RS=TP3S(
2))+(RZ=TP3Z(3))*(RS=TP3S(3)
)THEN 359Ø
387Ø RETURN
388Ø IF (RZ=TP4Z(1))*(RS=TP4
S(1))+(RZ=TP4Z(2))*(RS=TP4S(
2))+(RZ=TP4Z(3))*(RS=TP4S(3)
)THEN 359Ø
389Ø RETURN
39ØØ GOSUB 526Ø
391Ø RETURN
392Ø REM
393Ø REM
394Ø DOORZ=INT(RND*23)+1
395Ø IF DOORZ=6 THEN 393Ø
396Ø IF DOORZ=8 THEN 393Ø
397Ø IF DOORZ<6 THEN 398Ø EL
SE 4ØØØ
398Ø DOORS=7
399Ø GOTO 425Ø
4ØØØ IF DOORZ=7 THEN 4Ø1Ø EL
SE 4Ø3Ø
4Ø1Ø DOORS=8
4Ø2Ø GOTO 425Ø
4Ø3Ø IF (DOORZ>8)*(DOORZ<12)
THEN 4Ø4Ø ELSE 4Ø6Ø
4Ø4Ø DOORS=9
4Ø5Ø GOTO 425Ø
4Ø6Ø IF DOORZ=12 THEN 4Ø7Ø E
LSE 411Ø
4Ø7Ø REM
4Ø8Ø DOORS=INT(RND*11)+1Ø
4Ø9Ø IF (DOORS>9)*(DOORS<12)
THEN 41ØØ ELSE 4Ø7Ø
41ØØ GOTO 425Ø
411Ø IF DOORZ<18 THEN 412Ø E
LSE 414Ø
412Ø DOORS=12
413Ø GOTO 425Ø
414Ø IF DOORZ=18 THEN 415Ø E
LSE 419Ø
415Ø REM
416Ø DOORS=INT(RND*18)+13
417Ø IF (DOORS>12)*(DOORS<19
)THEN 418Ø ELSE 415Ø
418Ø GOTO 425Ø
419Ø IF DOORZ<24 THEN 42ØØ E
LSE 32767
42ØØ DOORS=19
421Ø GOTO 425Ø
422Ø IF (DOORZ=12)*(DOORS=9)
+(DOORZ=12)*(DOORS=12)THEN 3
93Ø
423Ø IF (DOORZ=18)*(DOORS=12
)+(DOORZ=18)*(DOORS=19)THEN
393Ø
424Ø GOTO 393Ø
425Ø REM
426Ø REM
427Ø MINED=INT(RND*2)+1
428Ø RETURN
429Ø REM
43ØØ RANDOMIZE
431Ø FOR M=1 TO 5
432Ø MI1Z(M)=INT(RND*11)+1
433Ø MI1S(M)=INT((1Ø-8+1)*RN
D)+8
434Ø NEXT M
435Ø FOR N=1 TO 5
436Ø MI2Z(N)=INT((17-11+1)*R
ND)+11
437Ø MI2S(N)=INT((15-11+1)*R
ND)+11
438Ø NEXT N
439Ø FOR O=1 TO 5
44ØØ MI3Z(O)=INT((23-17+1)*R
```

LISTINGS

```
ND)+17
4410 MI3S(O)=INT((20-16+1)*R
ND)+16
4420 NEXT O
4430 RETURN
4440 REM
4450 RANDOMIZE
4460 FOR P=1 TO 3
4470 TP1Z(P)=INT(RND*8)+1
4480 TP1S(P)=INT((15-11+1)*R
ND)+11
4490 NEXT P
4500 FOR Q=1 TO 3
4510 TP2Z(Q)=INT((14-7+1)*RN
D)+7
4520 TP2S(Q)=INT((20-16+1)*R
ND)+16
4530 NEXT Q
4540 FOR R=1 TO 3
4550 TP3Z(R)=INT((23-11+1)*R
ND)+11
4560 TP3S(R)=INT((25-21+1)*R
ND)+21
4570 NEXT R
4580 FOR T=1 TO 3
4590 TP4Z(T)=INT((23-18+1)*R
ND)+18
4600 TP4S(T)=INT((30-26+1)*R
ND)+26
4610 NEXT T
4620 RETURN
4630 REM
4640 EX=EX+1
4650 IF EX=8 THEN 4700
4660 TANKS(EX)=EX*2+14
4670 TANKZ(EX)=EX*2+2
4680 CALL HCHAR(TANKZ(EX),TA
NKS(EX),89)
4690 RETURN
4700 B$="dead"
4710 ZZ=14
4720 SS=6
4730 GOSUB 1410
4740 GOTO 4940
4750 REM
4760 IF EX<1 THEN 4930
4770 REM
4780 IF (RZ=TANKZ(1))+(RZ=TA
NKZ(2))+(RZ=TANKZ(3))+(RZ=TA
NKZ(4))+(RZ=TANKZ(5))+(RZ=TA
NKZ(6))THEN 4830
4790 IF (RZ=TANKZ(7))THEN 48
30
4800 IF (RS=TANKS(1))+(RS=TA
NKS(2))+(RS=TANKS(3))+(RS=TA
NKS(4))+(RS=TANKS(5))+(RS=TA
NKS(6))THEN 4920
4810 IF (RS=TANKS(7))THEN 49
20
4820 RETURN
```

```
4830 FOR I=30 TO 0 STEP -3.7
5
4840 CALL SOUND(100,-6,I)
4850 NEXT I
4860 CALL HCHAR(RZ,RS,34)
4870 RZ=10
4880 RS=26
4890 CALL HCHAR(RZ,RS,43)
4900 GOTO 4630
4910 RETURN
4920 GOTO 4830
4930 RETURN
4940 REM
4950 IF A$="dead" THEN 4979
4960 IF (HS>PTS)+(HS=0)THEN
4970 ELSE 4980
4970 HS=PTS
4979 CALL COLOR(14,1,1)
4980 CALL KEY(0,K,S)
4990 IF S=0 THEN 4980
5000 IF K=46 THEN 5010 ELSE
4980
5010 PTS=0
5020 ZAE=0
5030 RX=0
5040 EX=0
5050 FOR Y=1 TO 8
5060 TANKZ(Y)=0
5070 TANKS(Y)=0
5080 NEXT Y
5090 GOTO 150
5100 REM
5110 GOTO 5170
5120 CALL HCHAR(5,21,136)
5130 CALL HCHAR(6,21,137)
5140 CALL HCHAR(11,29,136)
5150 CALL HCHAR(12,29,137)
5160 RETURN
5170 GOSUB 5120
5180 CALL SCREEN(5)
5190 CALL COLOR(3,16,1)
5200 CALL COLOR(4,16,1)
5210 FOR I=9 TO 12
5220 CALL COLOR(I,16,1)
5230 NEXT I
5240 CALL COLOR(14,11,1)
5250 RETURN
5260 FOR I=0 TO 30 STEP 10
5270 CALL SOUND(100,-4,0)
5280 NEXT I
5290 RETURN
5300 FOR J=1 TO 3
5310 FOR I=220 TO 328 STEP 3
5320 CALL SOUND(-100,I,0)
5330 NEXT I
5340 FOR I=328 TO 220 STEP -
3
5350 CALL SOUND(-100,I,0)
5360 NEXT I
5370 NEXT J
```

```
5380 CALL COLOR(14,11,1)
5384 CALL COLOR(13,2,11)
5390 GOSUB 5120
5400 RETURN
5410 REM
```

STEUERUNG

RUN NORTH	E/JST2
RUN SOUTH	X/JST2
RUN WEST	S/JST2
RUN EAST	D/JST2
WURF	ENTER/FIRE
NEW TRY	>

ASCII DEF

CHAR 33	FOOTSTEPS
CHAR 34	EXPLOSION
CHAR 36–38/128	ASYL
CHAR 40	STACHELDRAHT
CHAR 41/42	BEOBACHTUNGSTURM
CHAR 43	RUNNER
CHAR 64	TODESSTREIFEN
CHAR 72	SICHTSTREIFEN
CHAR 80	MINENSPERRE
CHAR 88	SICHERUNGSSTREIFEN
CHAR 89	TANKS
CHAR 136/137	SUCHSCHEINWERFER
I J Y	ZÄHLWERTVARIABLEN
D	ZEITVARIABLE
M N O P Q R T	ORDNUNGSVARIABLEN
A$ B$ ZZ SS	DISPLAY ZEILE/SPALTE
RZ RS	RUNNER ZEILE/SPALTE
RX	RUNNER ZÄHLWERT
EX	PANZER ZÄHLWERT
ZAE	ZÄHLWERT
WURF	WURFWERT
PTS	PUNKTWERT
HS	HISCOREWERT
X Y	JOYSTICKWERTE
K S	KEYBOARDWERTE
SOU	SOUNDDEFINITION
MINED	TRETMINE
DOORZ DOORS	ÖFFNUNG ZEILE/SPALTE
TANKZ TANKS	PANZER ZEILE/SPALTE
TANK“(I)–TANK“(8)	PANZER
MI1“(X)–MI3“(X)	MINEN
TP1“(X)–TP4“(X)	MINEN
MI1Z–MI3Z	MINEN ZEILE
TP1Z–TP4Z	MINEN ZEILE
MI1S–MI3S	MINEN SPALTE
TP1S–TP4S	MINEN SPALTE

ADRESSKARTEI

„Adresskartei“ ist ein Adressenverwaltungsprogramm für den TI 99/4A, Ext. Basic, Disk und (optional) Drucker. Es kann bis zu einer vollen Diskette Adressen verwalten, was für die meisten Leser wohl völlig ausreichend ist.
Im Menü kann man 7 Optionen anwählen, nämlich: SUCHE, EINTRAG, REGISTER, ÄNDERN, LISTE, LÖSCHEN und ENDE.
SUCHE:
Einfach den gewünschten Namen eingeben und die Suche auf der Diskette beginnt. Sind mehrere Adressen mit gleichen Namen gespeichert, werden diese nacheinander ausgegeben.

EINTRAG: Bis zu 25 Einträge können nacheinander eingegeben werden, dann allerdings werden diese automatisch auf Diskette geschrieben, um den Arbeitsspeicher des TI's nicht zu überlasten. Dieser Vorgang benötigt eine gewisse Zeit, da der TI die Adressen alphabetisch sortiert. Dies ermöglicht dann sowohl eine schnelle Suche als auch einen schnellen Listenausdruck.



REGISTER: Einfach einen Buchstaben eingeben. Nacheinander erscheinen dann alle Namen mit dem entsprechenden Anfangsbuchstaben und zugehöriger Adresse.

ÄNDERN: Problemlos lassen sich in diesem Modus bestehende Einträge verändern.

LISTE: Die gesamte Adreßkartei wird (alphabetisch geordnet) über PIO ausgegeben.

LÖSCHEN: Einträge können so leicht vernichtet werden.

ENDE: Das Programm wird verlassen.

Vor dem ersten Programmstart muß zunächst eine Datei auf einer Diskette initialisiert werden. Das geschieht im Direktmodus auf folgende Weise:

```
OPEN #1:"DSK1.ADRESSEN",OUTPUT,
        RELATIVE, INTERNAL, FIXED 120
        <ENTER>
PRINT#1,REC 0:0  <ENTER>
CLOSE#1          <ENTER>
```

Es empfiehlt sich, das Programm unter dem Namen LOAD zusammen mit der Datei auf einer Diskette abzuspeichern, damit es dann auch wirklich schneller geht als mit dem Telefonbuch.

Andreas Finke

```
10 ! *********************
12 ! *    ADRESSKARTEI    *
14 ! *                    *
16 ! *    COPYRIGHT BY    *
18 ! *                    *
20 ! *   ANDREAS FINKE    *
22 ! *                    *
24 ! * BENOETIGTE GERAETE *
26 ! *  TI-99/4A KONSOLE  *
28 ! *     EXT. BASIC     *
30 ! *   1 DISKLAUFWERK   *
32 ! *                    *
34 ! *       OPT.:        *
36 ! *      DRUCKER       *
38 ! *                    *
40 ! *  SPEICHERBELEGUNG  *
42 ! *     5702 BYTES     *
44 ! *                    *
46 ! *********************
48 !
50 !
52 ! ACHTUNG !!!
54 ! ======= ===
56 ! Vor dem ersten Pro-
58 ! grammstart bitte im
60 ! Direktmodus eingeben:
62 ! OPEN #1:"DSK1.ADRESSEN
64 ! ",OUTPUT,RELATIVE,INTE
66 ! RNAL,FIXED 120 <ENTER>
68 ! PRINT #1,REC 0:0
70 ! <ENTER>
72 ! CLOSE #1  <ENTER>
74 !
76 !
78 OPTION BASE 1
80 CALL SCREEN(2):: FOR I=0
TO 14 :: CALL COLOR(I,11,2):
: NEXT I
82 DIM NA$(25),VN$(25),ST$(2
5),OR$(25),TE$(25)
84 !
86 ! MENUE
88 ! =====
90 !
92 CALL CLEAR :: DISPLAY AT(
6,9):"SUCHE     <1>" :: DISPL
AY AT(8,9):"EINTRAG  <2>" ::
 DISPLAY AT(10,9):"REGISTER
<3>" :: DISPLAY AT(12,9):"AE
NDERN  <4>"
94 DISPLAY AT(14,9):"LISTE
   <5>" :: DISPLAY AT(16,9):"
LOESCHEN <6>" :: DISPLAY AT(
18,9):"ENDE      <7>"
96 CALL KEY(0,K,S):: IF S=0
THEN 96 ELSE IF K<49 OR K>55
 THEN 96
98 CALL CLEAR :: ON K-48 GOT
O 108,118,154,166,178,202,20
8
100 !
102 ! SUCHE
104 ! =====
106 !
108 CALL NAME(NAME$):: CALL
LINSUCH(NAME$,1):: GOTO 92
110 !
112 ! EINTRAG
114 ! =======
116 !
118 FOR Z=1 TO 25 :: CALL CL
EAR
120 DISPLAY AT(1,12):"EINTRA
G" :: DISPLAY AT(2,10):RPT$(
"*",11)
122 DISPLAY AT(4,3):"NAME ?"
 :: DISPLAY AT(8,3):"VORNAME
 ?" :: DISPLAY AT(12,3):"STR
ASSE ?" :: DISPLAY AT(16,3):
"PLZ ORT ?" :: DISPLAY AT(20
,3):"TELEFONNR. ?"
124 ACCEPT AT(6,6)SIZE(-24):
NA$(Z):: ACCEPT AT(10,6)SIZE
(-24):VN$(Z):: ACCEPT AT(14,
6)SIZE(-24):ST$(Z):: ACCEPT
AT(18,6)SIZE(-24):OR$(Z)
126 ACCEPT AT(22,6)SIZE(-24)
:TE$(Z)
128 DISPLAY AT(23,3):"AENDER
N <A> WEITER <ENTER>"
130 CALL KEY(0,K,S):: IF S=0
 THEN 130 ELSE IF K=65 THEN
DISPLAY AT(23,3):" " :: GOTO
 124 ELSE IF K<>13 THEN 130
132 IF Z=25 THEN DISPLAY AT(
23,3)ERASE ALL:"AUTOMATISCHE
R EINTRAG" :: GOTO 140
134 DISPLAY AT(23,3):"FERTIG
 <F> WEITER <ENTER>"
136 CALL KEY(0,K,S):: IF S=0
 THEN 136 ELSE IF K=70 THEN
DISPLAY AT(23,3)ERASE ALL:"E
INTRAG ERFOLGT" :: GOTO 140
ELSE IF K<>13 THEN 136
138 NEXT Z
140 OPEN #1:"DSK1.ADRESSEN",
INTERNAL,UPDATE,RELATIVE,FIX
ED 120
142 INPUT #1,REC 0:ANZ :: FO
R J=1 TO Z :: CALL BUBBLE(NA
$(J),VN$(J),ST$(J),OR$(J),TE
$(J),ANZ+J-1):: NEXT J
144 PRINT #1,REC 0:ANZ+Z ::
CLOSE #1 :: GOTO 92
146 !
148 !  REGISTER
150 !  ========
152 !
154 DISPLAY AT(1,12):"REGIST
ER" :: DISPLAY AT(2,10):RPT$
```

```
("*",12):: DISPLAY AT(8,3):"
BUCHSTABE ?" :: ACCEPT AT(8,
16)VALIDATE(UALPHA)SIZE(1):B
U$
156 DISPLAY AT(23,3):"WIRD G
ESUCHT" :: CALL LINSUCH(BU$,
2):: GOTO 92
158 !
16Ø !  AENDERN
162 !  =======
164 !
166 DISPLAY AT(1,12):"AENDER
N" :: DISPLAY AT(2,1Ø):RPT$(
"*",11)
168 CALL NAME(NAME$):: CALL
LINSUCH(NAME$,3):: GOTO 92
17Ø !
172 !  LISTE
174 !  =====
176 !
178 DISPLAY AT(23,3):"AUSDRU
CK LAEUFT"
18Ø OPEN #1:"DSK1.ADRESSEN",
SEQUENTIAL,INTERNAL,INPUT ,F
IXED 12Ø
182 INPUT #1:ANZ :: IF ANZ=Ø
 THEN 192
184 OPEN #2:"PIO"
186 PRINT #2:CHR$(27);CHR$(4
5);CHR$(1);TAB(3Ø);"ADRESSEN
LISTE";CHR$(27);CHR$(45);CHR
$(Ø): : :
188 FOR I=1 TO ANZ :: INPUT
#1:A1$,A2$,A3$,A4$,A5$ :: PR
INT #2:A1$;"  ";A2$;"  ";A3$
;"  ";A4$;"  ";A5$
19Ø NEXT I :: CLOSE #2
192 CLOSE #1 :: GOTO 92
194 !
196 !  LOESCHEN
198 !  ========
2ØØ !
2Ø2 DISPLAY AT(1,12):"LOESCH
EN" :: DISPLAY AT(2,1Ø):RPT$
("*",12)
2Ø4 CALL NAME(NAME$):: CALL
LINSUCH(NAME$,4):: GOTO 92
2Ø6 !
2Ø8 END
21Ø !
212 !
214 !
216 ! UNTERPROGRAMME
218 ! ==============
22Ø !
222 SUB BUBBLE(A1$,A2$,A3$,A
4$,A5$,A)
224 IF A=Ø THEN PRINT #1,REC
 1:A1$,A2$,A3$,A4$,A5$ :: SU
BEXIT
```

LISTINGS

```
226 FOR K=A TO 1 STEP -1 ::
INPUT #1,REC K:B1$,B2$,B3$,B
4$,B5$
228 CALL UALPHA(A1$,B1$,TR):
: IF TR=1 THEN PRINT #1,REC
K+1:B1$,B2$,B3$,B4$,B5$ ELSE
 234
23Ø NEXT K
232 K=Ø
234 PRINT #1,REC K+1:A1$,A2$
,A3$,A4$,A5$
236 SUBEND
238 !
24Ø !
242 SUB UALPHA(A$,B$,TR)
244 MI=MIN(LEN(A$),LEN(B$)):
: FOR J=1 TO MI :: S1=ASC(SE
G$(A$,J,1)):: S2=ASC(SEG$(B$
,J,1))
246 IF S1<S2 THEN TR=1 :: SU
BEXIT
248 IF S1>S2 THEN TR=Ø :: SU
BEXIT
25Ø NEXT J
252 IF LEN(A$)<LEN(B$)THEN T
R=1 ELSE TR=Ø
254 SUBEND
256 !
258 !
26Ø SUB LINSUCH(A$,J)
262 FL=Ø
264 OPEN #1:"DSK1.ADRESSEN",
INTERNAL,FIXED 12Ø,SEQUENTIA
L,INPUT
266 INPUT #1:ANZ :: IF ANZ=Ø
 THEN SUBEXIT
268 FOR I=1 TO ANZ
27Ø INPUT #1:A1$,A2$,A3$,A4$
,A5$ :: IF J=2 THEN B$=SEG$(
A1$,1,1)ELSE B$=A1$
272 IF B$=A$ THEN CALL CLEAR
 :: DISPLAY AT(8,6):A1$ :: D
ISPLAY AT(1Ø,6):A2$ :: DISPL
AY AT(12,6):A3$ :: DISPLAY A
T(14,6):A4$ :: DISPLAY AT(16
,6):A5$ ELSE 286
274 FL=1 :: IF J=3 THEN 278
ELSE IF J=4 THEN 282 ELSE DI
SPLAY AT(23,3):"TASTE"
276 CALL KEY(Ø,K,S):: IF S=Ø
 THEN 276 ELSE 29Ø
278 DISPLAY AT(23,3):"AENDER
N <A> WEITER <ENTER>"
28Ø CALL KEY(Ø,K,S):: IF S=Ø
 THEN 28Ø ELSE IF K=65 THEN
CLOSE #1 :: DISPLAY AT(23,3)
:"WIRD GEAENDERT" :: GOTO 29
6 ELSE IF K=13 THEN 29Ø ELSE
 28Ø
282 DISPLAY AT(23,2):"LOESCH
```

```
EN <L> WEITER <ENTER>"
284 CALL KEY(Ø,K,S):: IF S=Ø
 THEN 284 ELSE IF K=76 THEN
CLOSE #1 :: DISPLAY AT(23,2)
:" WIRD GELOESCHT" :: GOTO 3
Ø6 ELSE IF K=13 THEN 29Ø ELS
E 284
286 IF FL=1 THEN FL=Ø :: DIS
PLAY AT(23,3):"TASTE" ELSE 2
9Ø
288 CALL KEY(Ø,K,S):: IF S=Ø
 THEN 288 ELSE CLOSE #1 :: S
UBEXIT
29Ø NEXT I :: IF FL=Ø THEN D
ISPLAY AT(23,3):"NICHT GEFUN
DEN" ELSE 294
292 FOR P=1 TO 4ØØ :: NEXT P
294 CLOSE #1 :: SUBEXIT
296 ACCEPT AT(8,6)SIZE(-24):
B1$ :: ACCEPT AT(1Ø,6)SIZE(-
24):B2$ :: ACCEPT AT(12,6)SI
ZE(-24):B3$ :: ACCEPT AT(14,
6)SIZE(-24):B4$ :: ACCEPT AT
(16,6)SIZE(-24):B5$
298 OPEN #1:"DSK1.ADRESSEN",
INTERNAL,RELATIVE,FIXED 12Ø,
UPDATE
3ØØ IF A$=B1$ THEN PRINT #1,
REC I:B1$,B2$,B3$,B4$,B5$ ::
 GOTO 31Ø
3Ø2 CALL LOESCH(I,ANZ):: CAL
L BUBBLE(B1$,B2$,B3$,B4$,B5$
,ANZ-1):: PRINT #1,REC Ø:ANZ
3Ø4 CLOSE #1 :: SUBEXIT
3Ø6 OPEN #1:"DSK1.ADRESSEN",
INTERNAL,RELATIVE,FIXED 12Ø,
UPDATE
3Ø8 CALL LOESCH(I,ANZ)
31Ø CLOSE #1 :: SUBEND
312 !
314 !
316 SUB LOESCH(K,A)
318 FOR I=K+1 TO A
32Ø INPUT #1,REC I:A1$,A2$,A
3$,A4$,A5$
322 PRINT #1,REC I-1:A1$,A2$
,A3$,A4$,A5$
324 NEXT I :: PRINT #1,REC Ø
:A-1
326 SUBEND
328 !
33Ø !
332 SUB NAME(NAME$)
334 DISPLAY AT(8,3):"NAME ?"
 :: ACCEPT AT(1Ø,3)SIZE(24):
NAME$ :: DISPLAY AT(23,3):"W
IRD GESUCHT"
336 SUBEND
```

SPACE-SHUTTLE FLUG-SIMULATOR

Aufgabe des Piloten ist es, das aus der Umlaufbahn zurückgekehrte Shuttle wieder sicher zu landen. Nach Aufbau des Cockpits startet das Spiel nach Drücken einer beliebigen Taste.

Funktionen:

HÖHE:
Die Höhe wird zweifach angezeigt, analog und digital.
In der analogen Anzeige zeigt ein gelber Balken die momentane Höhe an. Der zusätzliche blaue Balken zeigt an, in welcher Höhe man sich in 10 Sekunden befindet.

SINKG.:
Die Sinkgeschwindigkeit wird unterhalb der Höhe angezeigt und sollte beim Aufsetzen auf der Landebahn nicht größer als 1 meter/sec. sein.

GESCH.:
Rechts oben wird die Fluggeschwindigkeit angezeigt. Sie sollte beim Aufsetzen kleiner als 1000 sein. Sinkt sie unter 500, so besteht Absturzgefahr durch Überziehen.

DIST.:
Unterhalb der Geschwindigkeit wird die Distanz zum Landebahnanfang angezeigt.

DELTA:
Delta ist die absolute Abweichung vom Idealanflug. Es herrscht leichter Seitenwind.

RUDER:
Rechts unten wird die Stellung des Höhenruders angezeigt, in der Mitte unten die Stellung des Seitenruders.

FLAPS:
Die Landeklappen lassen sich erst bei einer Geschwindigkeit kleiner als 1500 ausfahren.

WHEELS:
Das Fahrwerk läßt sich erst bei einer Geschwindigkeit kleiner als 2000 ausfahren.

AUTOP.:
Der Autopilot landet das Shuttle fast immer sicher.

STEUERUNG:
- A = 3 Grad nach links
- S = 1 Grad nach links
- D = 1 Grad nach rechts
- F = 3 Grad nach rechts
- G = Ein-/Ausfahren der Landeklappen
- W = Ein-/Ausfahren des Fahrwerks
- ENTER = Ein-/Ausschalten des Autopiloten
- B = Bremsen auf der Landebahn

LISTINGS

```
10 ! *********************
11 ! *                   *
12 ! * SHUTTLE SIMULATOR *
13 ! *                   *
14 ! *    Copyright by   *
15 ! * Michael Charisse' *
16 ! *                   *
17 ! *                   *
19 ! * Benoetigte Geraete *
20 ! *  TI99/4A  Konsole *
21 ! *     Ext. Basic    *
22 ! *                   *
26 ! *  Speicherbelegung *
27 ! *     8794 Bytes    *
28 ! *                   *
29 ! *********************
30 !
230 ! KURZANLEITUNG :
240 ! LINKS  =  A S
250 ! RECHTS =  D F
260 ! HOCH   =  E
270 ! RUNTER =  X
280 ! LANDEKLAPPEN = G
290 ! FAHRWERK     = W
300 ! AUTOPILOT    = ENTER
310 ! REIFENBREMSE = B
320 !
350 ON ERROR 2600
360 !
370 CALL CLEAR :: CALL DELSP
RITE(ALL):: CALL CHARSET ::
RANDOMIZE
380 CALL SCREEN(5)
390 CALL MAGNIFY(4)
400 FOR I=1 TO 14 :: CALL CO
LOR(I,16,2)
410 NEXT I
420 DIM C(250)
430 ! COCKPIT-GRAFIK
440 CALL CHAR(136,"FF8080808
0808080")
450 CALL CHAR(137,"808080808
08080FF")
460 CALL CHAR(138,"010101010
10101FF")
470 CALL CHAR(139,"FF0101010
1010101")
480 CALL CHAR(140,"FF0000000
0000000")
495 CALL CHAR(141,"808080808
0808080")
500 CALL CHAR(142,"000000000
00000FF")
510 CALL CHAR(143,"010101010
1010101")
520 CALL CHAR(128,"101010101
0101010")
530 CALL CHAR(129,"FFFFFFFFF
FFFFFFF")
540 CALL CHAR(120,"FF81C3818
181C381")
550 CALL CHAR(121,"8181C3818
181C381")
560 CALL CHAR(122,"8181C3818
18181FF")
570 CALL CHAR(112,"FFFFFFFFF
FFFFFFF")
580 CALL CHAR(113,"FF8181818
18181FF")
590 ! SPRITES-GRAFIK
600 W0$="0000000010101037F0
00000000000000000000000008
0FE0000000000000"
610 W1$="0000000010101037F0
80800000000000000000000008
0FE2020000000000"
620 CALL CHAR(104,"7"&RPT$("
0",63))
630 CALL CHAR(108,"000000100
0000000"&RPT$("0",48))
640 CALL CHAR(96,"8080808080
8080800000000000000000"&RPT$
("0",32))
650 CALL CHAR(92,"0010103810
000000"&RPT$("0",48))
660 ! COCKPIT-AUFBAU
670 CALL CHAR(100,W0$)
680 CALL COLOR(14,6,2)
690 CALL HCHAR(1,12,136)
700 CALL VCHAR(2,12,141,17)
710 CALL HCHAR(19,12,137)
720 CALL HCHAR(19,13,142,7)
730 CALL HCHAR(19,20,138)
740 CALL VCHAR(2,20,143,17)
750 CALL HCHAR(1,20,139)
760 CALL HCHAR(1,13,140,7)
770 CALL COLOR(13,16,2)
780 CALL HCHAR(2,16,128)
790 CALL HCHAR(21,13,140,7)
800 CALL HCHAR(23,13,142,7)
810 CALL HCHAR(21,12,136)
820 CALL HCHAR(22,12,141)
830 CALL HCHAR(23,12,137)
840 CALL HCHAR(21,20,139)
850 CALL HCHAR(22,20,143)
860 CALL HCHAR(23,20,138)
870 CALL HCHAR(22,13,128,7)
880 CALL COLOR(12,16,2)
890 CALL HCHAR(1,2,120)
900 CALL VCHAR(2,2,121,22)
910 CALL HCHAR(24,2,122)
920 DISPLAY AT(1,4)SIZE(3):"
ALT"
930 DISPLAY AT(7,3)SIZE(5):"
VARIO"
940 DISPLAY AT(14,3)SIZE(5):
"FLAPS"
950 DISPLAY AT(20,3)SIZE(6):
"WHEELS"
960 CALL COLOR(11,12,5)
```

LISTINGS

```
97Ø CALL HCHAR(16,6,112)
98Ø CALL HCHAR(16,7,113)
99Ø CALL HCHAR(22,6,112)
1ØØØ CALL HCHAR(22,7,113)
1Ø1Ø CALL HCHAR(18,26,14Ø,2)
1Ø2Ø CALL HCHAR(24,26,14Ø,2)
1Ø3Ø CALL HCHAR(19,25,14Ø)::
 CALL HCHAR(19,28,14Ø)
1Ø4Ø CALL HCHAR(23,25,14Ø)::
 CALL HCHAR(23,28,14Ø)
1Ø5Ø CALL HCHAR(2Ø,24,14Ø,2)
:: CALL HCHAR(22,24,14Ø,2)
1Ø6Ø CALL HCHAR(2Ø,28,14Ø,2)
:: CALL HCHAR(22,28,14Ø,2)
1Ø7Ø CALL HCHAR(21,23,14Ø,3)
:: CALL HCHAR(21,28,14Ø,3)
1Ø8Ø DISPLAY AT(1,22)SIZE(5)
:"SPEED"
1Ø9Ø DISPLAY AT(7,22)SIZE(4)
:"DIST"
11ØØ DISPLAY AT(13,22)SIZE(5
):"DELTA"
111Ø ! SPRITES POSITIONIEREN
112Ø CALL SPRITE(#1,1ØØ,12,1
44,194)
113Ø CALL SPRITE(#2,1Ø4,12,1
,8)
114Ø CALL SPRITE(#6,1Ø4,5,1,
8)
115Ø CALL SPRITE(#3,1Ø8,16,1
5,124)
116Ø CALL SPRITE(#4,96,7,165
,124)
117Ø ! FLUGPARAMETER
118Ø V=3ØØØ
119Ø AUF,LAN,FA=Ø
12ØØ WI=Ø
121Ø FL,WH,AP=Ø
122Ø CO=Ø
123Ø DX=INT(RND*3Ø)+17Ø
124Ø DY=RND*6Ø-3Ø
125Ø AL,ALØ=INT(RND*5ØØØ)+8Ø
ØØ
126Ø BL=(RND-.5)/3Ø
127Ø GOSUB 166Ø
128Ø ! HIER HAUPTPROGRAMM
129Ø CALL KEY(Ø,K,S):: IF S=
Ø THEN 129Ø
13ØØ GOSUB 151Ø
131Ø IF AP=1 THEN GOSUB 179Ø
132Ø IF V>3ØØØ OR V<6ØØ THEN
 DISPLAY AT(1,22)SIZE(5):"pp
ppp" :: DISPLAY AT(1,22)SIZE
(5):"SPEED"
133Ø IF AUF<-4Ø OR AUF>Ø THE
N DISPLAY AT(7,3)SIZE(5):"pp
ppp" :: DISPLAY AT(7,3)SIZE(
5):"VARIO" :: DISPLAY AT(7,3
)SIZE(5):"ppppp" :: DISPLAY
AT(7,3)SIZE(5):"VARIO"
134Ø C(5Ø+DX)=AL
135Ø AUF=(WI+FL-.1)*(5+FL*2*
SGN(WI))*(V-15Ø)/5ØØØ
136Ø IF V<5ØØ THEN AUF=AUF-(
(55Ø-V)/5Ø)^4
137Ø V=V-((V/2ØØØ)^2)*(5Ø+WH
*25+FL*15)
138Ø V=V-((V+1ØØØ)/3Ø)*SIN(2
*WI*PI/18Ø)
139Ø RV=COS(WI*PI/18Ø)*V
14ØØ DY=BL+DY-SIN(CO*PI/18Ø)
*RV/4ØØØ
141Ø DX=DX-COS(CO*PI/18Ø)*RV
/35ØØ
142Ø AL=AL+AUF
143Ø IF LAN=1 THEN GOSUB 2Ø5
Ø :: AL=MAX(AL,Ø):: AUF=MAX(
AUF,Ø):: IF AL>Ø THEN LAN=Ø
ELSE IF V<Ø THEN V=Ø :: GOTO
 214Ø
144Ø IF FA=Ø THEN GOSUB 166Ø
 ELSE GOSUB 192Ø
145Ø IF FA=Ø AND DX<15 AND D
Y*SGN(DY)<4 THEN CALL VCHAR(
2,16,129,9):: FA=1 :: CALL C
OLOR(#3,7):: CALL PATTERN(#3
,92):: GOSUB 21ØØ
146Ø IF AL>.1 THEN 13ØØ
147Ø IF AUF<-1 OR WH=Ø OR DX
<-5Ø OR DY*SGN(DY)>5 OR V>1Ø
ØØ THEN LAN=Ø :: GOTO 214Ø
148Ø CALL SOUND(-5Ø,111Ø,25)
:: LAN=1
149Ø GOTO 13ØØ
15ØØ ! TASTATUR-MODUL
151Ø CALL KEY(Ø,K,S):: IF S=
Ø THEN RETURN
152Ø CALL SOUND(-1ØØ,9ØØ,29,
9Ø5,29,91Ø,29)
153Ø IF K=69 THEN WI=WI+1 ::
 WI=MIN(3Ø,WI)
154Ø IF K=88 THEN WI=WI-1 ::
 WI=MAX(-3Ø,WI)
155Ø IF K=83 THEN CO=CO-1 ::
 CO=MAX(-34,CO)
156Ø IF K=65 THEN CO=CO-3 ::
 CO=MAX(-34,CO)
157Ø IF K=68 THEN CO=CO+1 ::
 CO=MIN(34,CO)
158Ø IF K=7Ø THEN CO=CO+3 ::
 CO=MIN(34,CO)
159Ø IF K=71 AND FL=Ø AND V<
15ØØ THEN FL=1 :: CALL HCHAR
(16,6,113):: CALL HCHAR(16,7
,112):: RETURN
16ØØ IF K=71 AND FL=1 THEN F
L=Ø :: CALL HCHAR(16,6,112):
: CALL HCHAR(16,7,113):: RET
URN
161Ø IF K=87 AND WH=Ø AND V<
```

LISTINGS

```
2000 THEN WH=1 :: CALL HCHAR
(22,6,113):: CALL HCHAR(22,7
,112):: CALL CHAR(100,W1$)::
 RETURN
1620 IF K=87 AND WH=1 THEN W
H=0 :: CALL HCHAR(22,6,112):
: CALL HCHAR(22,7,113):: CAL
L CHAR(100,W0$)
1630 IF K=13 AND AP=0 THEN A
P=1 ELSE AP=0 :: DISPLAY AT(
4,3)SIZE(5):"      " :: ! EIN
-/AUSSCHALTEN DES AUTOPILOTE
N
1640 RETURN
1650 ! ANZEIGE-MODUL
1660 DISPLAY AT(3,23)SIZE(4)
:USING "####":V
1670 DISPLAY AT(3,3)SIZE(5):
USING "#####":AL
1680 DISPLAY AT(9,2)SIZE(6):
USING "####.#":AUF
1690 DISPLAY AT(9,22)SIZE(5)
:USING "###.#":DX
1700 DISPLAY AT(15,22)SIZE(6
):USING "##.##":SGN(DY)*DY
1710 CALL LOCATE(#1,142-WI,1
93)
1720 CALL LOCATE(#2,191-AL/8
0,8)
1730 IF AUF<-100 THEN 1750
1740 CALL LOCATE(#6,191-(AL+
AUF*10)/80,8)
1750 CALL LOCATE(#3,9+DX/200
*135,117-DY)
1760 CALL LOCATE(#4,165,124+
CO)
1770 RETURN
1780 ! MODUL AUTOPILOT
1790 DISPLAY AT(13,12)SIZE(5
):"AUTO-" :: DISPLAY AT(15,1
2)SIZE(5):"PILOT"
1800 IF AL<200 AND DX<10 AND
 WH=0 THEN K=87 :: GOSUB 152
0
1810 IF DX>50 THEN SH=50*DX
ELSE IF DX>10 THEN SH=(DX-10
)*40 ELSE IF DX>0 THEN SH=10
 ELSE IF DX>-5 THEN SH=1 ELS
E SH=-1
1820 DH=SH-AL
1830 DH=DH/10
1840 IF AUF<DH THEN WI=WI+1
1850 IF AUF>DH AND WI>-35 TH
EN WI=WI-1
1860 DISPLAY AT(4,3)SIZE(5):
USING "#####":SH
1870 IF CO<8*DY THEN CO=CO+1
 :: CO=MIN(CO,34)
1880 IF CO>8*DY THEN CO=CO-1
 :: CO=MAX(CO,-34)
1890 DISPLAY AT(13,12)SIZE(5
):"     " :: DISPLAY AT(15,1
2)SIZE(5):"     "
1900 RETURN
1910 ! MODUL ANZEIGE FUER LA
NDEANFLUG
1920 DISPLAY AT(3,23)SIZE(4)
:USING "####":V
1930 DISPLAY AT(3,3)SIZE(6):
USING "####.#":AL
1940 DISPLAY AT(9,2)SIZE(6):
USING "####.#":AUF
1950 DISPLAY AT(9,22)SIZE(5)
:USING "###.#":DX*10
1960 DISPLAY AT(15,22)SIZE(6
):USING "###.##":SGN(DY)*DY*
10
1970 CALL LOCATE(#1,142-WI,1
93)
1980 IF AL>85 THEN CALL LOCA
TE(#2,191-AL/15,8)ELSE CALL
LOCATE(#2,191-AL*2,8)
1990 IF (AL+AUF*10)*2>191 TH
EN 2000
2000 CALL LOCATE(#3,76+DX*4.
5,118-DY*7)
2010 IF AL<120 THEN CALL LOC
ATE(#7,123-AL,110-DY*40)
2020 CALL LOCATE(#4,165,124+
CO)
2030 RETURN
2040 ! MODUL AUSROLLEN
2050 CALL KEY(0,K,S):: IF K=
66 THEN V=V-30
2060 IF V<400 THEN WI=0
2070 V=V*.95
2080 RETURN
2090 ! MODUL LANDEBAHN
2100 CALL HCHAR(19,12,129,9)
2110 CALL SPRITE(#7,100,15,1
23,110)
2120 RETURN
2130 ! FLUGBEWERTUNG
2140 FOR DEL=1 TO 300 :: NEX
T DEL
2150 CALL CLEAR
2160 CALL DELSPRITE(ALL)
2170 CALL HCHAR(24,1,142,32)
2180 CALL SPRITE(#2,96,7,185
,205)
2190 CALL SPRITE(#1,108,16,2
00,200)
2200 FOR I=250 TO 20 STEP -1
2210 CALL LOCATE(#1,184-C(I)
/82,251-I)
2220 NEXT I
2225 DISPLAY AT(3,1):"DRUECK
E TASTE FUER WEITER"
2230 FOR DE=1 TO 100 :: CALL
 KEY(0,K,S):: IF S<>0 THEN 2
```

```
260
2240 NEXT DE
2245 DISPLAY AT(3,1):"
                         "
2250 GOTO 2200
2260 CALL CLEAR :: CALL SCRE
EN(12)
2270 FOR I=1 TO 14 :: CALL C
OLOR(I,2,1):: NEXT I
2280 CALL DELSPRITE(ALL)
2290 IF LAN=1 THEN 2390
2300 DISPLAY AT(1,1):"ES TUT
 MIR SEHR LEID, ABER  SIE SI
ND ZERSCHELLT!"
2310 DISPLAY AT(3,1):"DIE TR
UEMMER DER ENTERPRISE LIEGEN
 ";INT(DX*10)/10:"MEILEN VOM
  AUFSETZPUNKT    ENTFERNT."
2320 DISPLAY AT(7,1):"SIE SI
ND IM UMKREIS VON ";INT(RND*
10)+2
2330 DISPLAY AT(8,1):"MEILEN
 VERSTREUT."
2340 DISPLAY AT(11,1):"SIE Z
ERSCHELLTEN MIT EINER  GESCH
WINDIGKEIT VON ";INT(V):"MEI
LEN PRO STUNDE."
2350 IF WH=0 THEN DISPLAY AT
(15,1):"DAS FAHRWERK WAR NIC
HT       AUSGEFAHREN"
2360 DISPLAY AT(22,1):"DRUEC
KE TASTE FUER WEITER"
2370 CALL KEY(0,K,S):: IF S=
0 THEN 2370
2380 GOTO 2500
2390 FOR I=1 TO 5 :: DISPLAY
 AT(1,1):"   G R A T U L A T
 I O N"
2400 FOR DE=1 TO 100 :: NEXT
 DE
2410 DISPLAY AT(1,1):"
                         "
2420 FOR DE=1 TO 100 :: NEXT
 DE
2430 NEXT I
2440 DISPLAY AT(1,1):"SIE SI
ND SICHER GELANDET."
2450 DISPLAY AT(3,1):"IHRE M
ASCHINE STEHT ";ABS(INT(DX*1
0)/10)
2460 DISPLAY AT(5,1):"MEILEN
 HINTER DEM ANFANG "
2470 DISPLAY AT(7,1):"DER ST
ARTBAHN. "
2480 DISPLAY AT(20,1):"DRUEC
KE TASTE FUER WEITER"
2490 CALL KEY(0,K,S):: IF S=
0 THEN 2490
2500 CALL CLEAR
2510 DISPLAY AT(12,1):"NOCH
EIN VERSUCH (J/N) ?"
2520 ACCEPT AT(12,26)VALIDAT
E("JN")BEEP:FR$
2530 IF FR$="N" THEN STOP
2540 CALL CLEAR
2550 CALL SCREEN(5)
2560 FOR I=1 TO 14 :: CALL C
OLOR(I,16,2):: NEXT I
2570 FOR I=1 TO 250 :: C(I)=
0 :: NEXT I
2580 GOTO 660
2600 DISPLAY AT(22,1):"
                         "
2605 DISPLAY AT(23,1):"    F
 L U G F E H L E R      "
2610 DISPLAY AT(24,1):" DRUE
CKE TASTE FUER NEUSTART"
2620 CALL KEY(0,K,S):: IF S=
0 THEN 2620
2630 DISPLAY AT(23,1):"

                         "
2640 RUN
2650 END
```

LISTINGS

MONSTER-ATTACK

Ähnlich aufgebaut wie der bekannte Atari-Oldie „Space Invaders", müßt Ihr bei diesem Spiel versuchen, Monster abzuschießen, die sich langsam und Zeile für Zeile dem unteren Bildschirmrand nähern. Im Gegensatz zu Space-Invaders schießen die merkwürdigen Monsterfratzen jedoch nicht direkt zurück, sondern verfügen über einen Killersatelliten, der gezielt Lenkraketen auf Eure Kanone abfeuert. Die Waffen sind mit einem elektronischen Feinderkennungssystem ausgestattet und machen auch jede Richtungsänderung Eurer eigenen Kanone mit.
Es ist damit nicht ganz einfach, dem perfekten Zerstörungssystem zu entkommen und nebenbei auch noch die vier Monster in jedem Bildschirm abzuschießen, zumal diese von Runde zu Runde schneller werden und schwieriger zu vernichten sind.
Gespielt wird mit Joystick #1. Als Bewegungsrichtung für die eigene Kanone ist nur der horizontale Weg (rechts–links) offen. Dabei darf der Bildschirm jedoch nach beiden Seiten verlassen werden und der Spieler kann den Lenkwaffen auf diese Weise öfters schon einmal entkommen.
Eigene Raketen werden abgefeuert, indem man die Kanone mit dem Joystick unter ein Monster oder in dessen Bewegungsrichtung steuert und ihn anschließend nach oben drückt. Damit wird eine Rakete gestartet. Zur Explosion kommt sie allerdings erst durch eine Gegenbewegung des Joysticks.
Wurde die Rakete in unmittelbarer Nähe eines Monsters zur Explosion gebracht, bedeutet dies einen Feind weniger auf der Spielfläche.
Gegnerische Lenkwaffen können nicht abgeschossen werden. Auch solltet Ihr beachten, daß sich Eure Kanone nur mit langsamer Geschwindigkeit steuern läßt, solange eines Eurer Geschosse unterwegs ist.
Eine leichte Beute für den Gegner???

LISTINGS

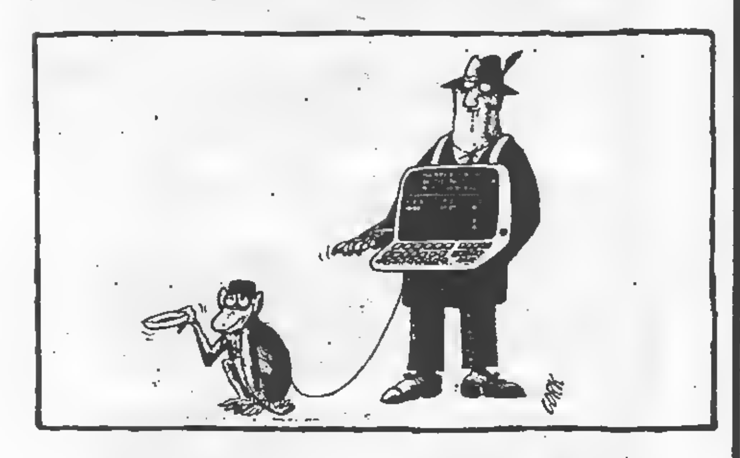
```
1Ø ! **********************
11 ! *                    *
12 ! *   MONSTER ATTACK   *
13 ! *                    *
14 ! *    Copyright by    *
15 ! *   Norbert  Frese   *
16 ! *                    *
19 ! * Benoetigte Geraete *
2Ø ! *  TI99/4A  Konsole  *
21 ! *     Ext. Basic     *
22 ! *     Joystick 1     *
23 ! *                    *
26 ! *  Speicherbelegung  *
27 ! *     5638 Bytes     *
28 ! *                    *
29 ! **********************
3Ø !
22Ø CALL MAGNIFY(3)
23Ø CALL CLEAR :: CALL SCREE
N(13):: CALL COLOR(1,16,2,2,
16,2,3,16,2,4,16,2,5,16,2,6,
16,2,7,16,2,8,16,2,9,13,13)
24Ø CALL HCHAR(17,1,96,32)
25Ø CALL CHAR(1ØØ,"3C48E69ØC
Ø8872202Ø7288CØ9ØE6483C3C126
7Ø9Ø3114EØ4Ø44E11Ø3Ø967123C"
)
26Ø CALL CHAR(1Ø4,"Ø1Ø1Ø1Ø
1Ø1Ø1Ø1Ø1Ø3Ø5Ø9Ø9Ø9ØF8Ø8Ø8
Ø8Ø8Ø8Ø8Ø8Ø8ØCØAØ9Ø9Ø9ØFØ"
)
27Ø CALL CHAR(1Ø8,"894522111
111111111113F4Ø8Ø8Ø8ØFF91A24
4888888888888888FCØ2Ø1Ø1Ø1FF"
)
28Ø CALL CHAR(112,"38Ø43A444
22E9ECCAØ9F88969FAØCØ8Ø1C2Ø5
C2242747933Ø5F99149F9Ø5Ø3Ø1"
)
29Ø CALL CHAR(116,"8ØCØA3949
89A5A3A18Ø9Ø9Ø8Ø8ØBØ7Ø3Ø1Ø3C
52919595A5C189Ø9Ø1Ø1ØDØEØCØ"
)
3ØØ CALL CHAR(35,"ØØ82C6EED6
C6C6")
31Ø CALL CHAR(36,"ØØ6Ø6Ø6Ø6Ø
667C")
32Ø CALL CHAR(37,"ØØ386CC6C6
FEC6")
33Ø CALL CHAR(38,"ØØFECØF8CØ
FE")
34Ø CALL CHAR(43,"ØØ7CC67Ø1C
C67C")
35Ø DISPLAY AT(3,1):"##### #
## ### ## ### ### #### # # #
 # # # #...#..#...# ## # # #
 # # # ### #..##..## # # # #
 # # #...# #..#...# #"
36Ø DISPLAY AT(7,1):"# # # #
## # # ### #..### # #"
37Ø DISPLAY AT(1Ø,1):"..%%..
%%%%%..%%...%%% %..%..%  %.
.%..%..%..%.%....% %...%%%%.
.%..%..%%%% %....%%....%..%
 %..%..%..% %....% %"
38Ø DISPLAY AT(14,1):" %..%
 %..%..%..%..%%% %..%"
39Ø DISPLAY AT(19,7):"VON NO
RBERT FRESE": :"<FEUERKNOPF>
______SPIELSTART": :"<A>____
____________ANLEITUNG"
4ØØ CALL KEY(1,C,S):: IF C=1
 THEN 93Ø
41Ø IF C<>18 THEN 4ØØ
42Ø CALL CLEAR
43Ø R=1 :: SC=Ø :: L=7
44Ø ON R GOSUB 86Ø,87Ø,88Ø,8
9Ø,9ØØ,91Ø,92Ø
45Ø CALL SPRITE(#1,1Ø8,16,17
7,5Ø,#3,112,16,2Ø,2ØØ,#4,112
,16,1Ø,1Ø)
46Ø CALL SPRITE(#5,112,16,1Ø
,1ØØ)
47Ø CALL SPRITE(#7,116,16,3Ø
,3Ø)
48Ø MO=3
49Ø GOSUB 5ØØ :: GOSUB 67Ø :
: GOSUB 5ØØ :: GOSUB 67Ø ::
GOSUB 5ØØ :: GOSUB 67Ø :: GO
SUB 62Ø :: GOSUB 64Ø :: GOTO
 49Ø
5ØØ CALL JOYST(1,JX,JY):: CA
LL MOTION(#1,Ø,JX*3):: IF JY
=4 THEN GOTO 51Ø ELSE RETURN
51Ø CALL POSITION(#1,S,B)::
CALL SPRITE(#2,1Ø4,16,155,B,
-23,JX*4)
52Ø FOR X=1 TO Ø9 :: CALL JO
YST(1,JX,JY)
53Ø CALL MOTION(#1,Ø,JX):: I
F JY=-4 THEN 55Ø
54Ø GOSUB 64Ø :: NEXT X :: C
ALL DELSPRITE(#2):: RETURN
55Ø CALL SOUND(1ØØ,-7,Ø):: C
ALL PATTERN(#2,1ØØ):: CALL M
OTION(#2,Ø,Ø)
56Ø FOR Y=3 TO 5 :: IF M(Y)T
HEN CALL COINC(#2,#Y,9,C)::
IF C THEN CALL PATTERN(#Y,1Ø
Ø):: CALL SOUND(1ØØ,-7,Ø)::
CALL SOUND(-5Ø,-5,Ø,1ØØØ,3):
: GOTO 59Ø
57Ø NEXT Y
58Ø CALL DELSPRITE(#2):: RET
URN
59Ø SC=SC+1 :: M(Y)=M(Y)-1 :
: IF M(Y)THEN CALL SOUND(1ØØ
,-1,Ø):: CALL PATTERN(#Y,112
):: GOTO 58Ø
6ØØ MO=MO-1 :: IF MO THEN CA
```

```
LL DELSPRITE(#Y):: GOTO 590
610 GOSUB 770 :: GOTO 440
620 FOR Y=3 TO 5 :: IF M(Y)T
HEN CALL MOTION(#Y,RND*1,RND
*MG-MG/2):: CALL POSITION(#Y
,C,S):: IF C>177 THEN GOTO 6
80
630 NEXT Y :: RETURN
640 CALL POSITION(#1,S,B,#7,
S,A)
650 IF B<A THEN V=-AG ELSE V
=AG
660 CALL MOTION(#7,7,V)
670 CALL COINC(#1,#7,16,C)::
 IF C THEN 680 ELSE RETURN
680 CALL PATTERN(#1,100):: C
ALL SOUND(100,-7,0,220,3)::
CALL SOUND(-100,-5,0,2000,0)
:: CALL SOUND(50,-6,0)
690 L=L-1 :: IF L>0 THEN CAL
L PATTERN(#1,108):: CALL DEL
SPRITE(#2):: RETURN
700 CALL DELSPRITE(ALL)
710 DISPLAY AT(2,1):"++++ ++
++ +++ ++++ +........+....+.
.+..+..+....+........++++ ++
++..+..+++..+...........+ +
....+..+....+"
720 DISPLAY AT(6,1):"++++ +.
...+++ ++++ ++++............
............................
..&&&& &..& &&&& &&&&.......
..&....&& &..& & &..."
730 DISPLAY AT(10,1):"......
...&&&..& &&..& & &&&.......
...&....&..&..& & &.........
...&&&& &..& &&&& &&&&"
740 DISPLAY AT(15,8):"TREFFE
R:";SC: :"WOLLEN SIE NOCHEIN
MAL  ????": :"DRUECKEN SIE
BITTE:": :"<FEUER>______FUER
 NOCHEINMAL": :"<E>_________
_FUER ENDE"
750 CALL KEY(1,C,S):: IF C=1
8 THEN CALL CLEAR :: CALL HC
HAR(17,1,96,32):: GOTO 350 E
LSE IF C<>5 THEN 750
760 CALL CLEAR :: END
770 DISPLAY AT(2,1):"$....$$
$$$ $...$ $$$$$ $....$....$.
....$...$ $.....$....$....$$
$$..$...$ $$$$..$....$....$.
.....$ $..$.....$"
780 DISPLAY AT(6,1):"$$$$ $$
$$$...$...$$$$$ $$$$"
790 CALL DELSPRITE(ALL):: DI
SPLAY AT(15,5):"GESCHAFFTER
LEVEL";R
800 DISPLAY AT(17,4):"TREFFE
R:";SC;"LEBEN";L
810 IF R=7 THEN DISPLAY AT(2
,1):"BRAVO !!! SIE HABEN ALL
E": :"RUNDEN GESCHAFFT !!!"
:: R=1 :: L=L+4
820 FOR X=220 TO 440 STEP 10
 :: CALL SOUND(50,X,0):: NEX
T X
830 R=R+1 :: DISPLAY AT(20,1
):"....FEUERKNOPF DRUECKEN..
.."
840 CALL KEY(1,C,S):: IF S=0
 THEN 840 ELSE CALL CLEAR ::
 RETURN
850 !@p-
860 M(3),M(4),M(5)=1 :: MG=1
8 :: AG=13 :: RETURN
870 M(3),M(4),M(5)=1 :: MG=1
8 :: AG=16 :: RETURN
880 M(3),M(4)=1 :: M(5)=2 ::
 MG=18 :: AG=16 :: RETURN
890 M(3),M(4),M(5)=2 :: MG=8
 :: AG=10 :: RETURN
900 M(3)=3 :: M(4),M(5)=1 ::
 MG=8 :: AG=12 :: RETURN
910 M(3)=6 :: M(4),M(5)=1 ::
 MG=6 :: AG=8 :: RETURN
920 M(3)=4 :: M(4),M(5)=2 ::
 MG=14 :: AG=14 :: RETURN
930 DISPLAY AT(2,1)ERASE ALL
:"------ SPIELANLEITUNG ----
--": :"MIT DEM JOYSTICK WIRD
 DIE": :"BASIS(UNTEN) NACH L
INKS UND"
940 DISPLAY AT(8,1):"NACH RE
CHTS BEWEGT. DRUECKT": :"MAN
 DEN JOYSTICK NACH": :"OBEN,
 SCHIESZT MAN EINE": :"GRANA
TE AB. MIT IHR KANN"
950 DISPLAY AT(16,1):"MAN EI
N 'LACHMONSTER' ZERST-": :"O
EREN WENN MAN SIE MIT": :"EI
NEM DRUCK NACH UNTEN MIT": :
"DEM JOYSTICK EXPLODIEREN- >
>"
960 CALL KEY(1,C,S):: IF C<>
18 THEN 960
970 DISPLAY AT(2,1)ERASE ALL
:"LAESST.": :"DIE ANGREIFEND
E GRANATE": :"LENKT IHNEN NA
CH UND SIE": :"KOENNEN SIE M
EIST NUR MIT"
980 DISPLAY AT(10,1):"TRICKS
 ABSCHUETTELN.": :"SIE BLEIB
T DIE GANZE ZEIT,": :"BIS AL
LE 'LACHMONSTER' BES-": :"EI
TIGT SIND."
990 DISPLAY AT(18,1):"MANCHE
 LACHMONSTER SIND SO": :"KRA
EFTIG, DASZ SIE EININIGE": :
"TREFFER DURCHHALTEN. >>"
```

```
1000 CALL KEY(1,C,S):: IF C<
>18 THEN 1000
1010 DISPLAY AT(2,1)ERASE AL
L:"DIE MONSTER SENKEN SICH":
 :"LANGSAM HERUNTER; WENN SI
E": :"DEN BODEN BERUEHRT HAB
EN": :"WIRD DIE BASIS ZERSTO
ERT."
1020 DISPLAY AT(10,1):"MAN K
ANN GRANATEN AUCH": :"SCHRAE
G ABFEUERN, WENN MAN": :"DEN
 JOYSTICK SCHRAEG NACH": :"O
BEN DRUECKT!"
1030 DISPLAY AT(18,1):"VORSI
CHT! WENN EINE GRANATE": :"U
NTERWEGS IST, KANN MAN NUR":
 :"LANGSAM LENKEN!-VIEL GLUE
CK>"
1040 CALL KEY(1,C,S):: IF C<
>18 THEN 1040 ELSE 420
1050 !@P+
```

DECKEL DRAUF

Deckel drauf ist ein Würfelspiel für eine Person. Im Spielfeld, einem bunten Bildschirmfenster, befinden sich in der oberen Hälfte neun numerierte Kästchen, die es eins nach dem anderen auszulöschen gilt.
Zum Löschen der Kästchen dienen zwei Würfel. Der Spieler kann entweder die Augenzahl eines Würfels oder die Summe beider Würfel zum Schließen der Nummernkästchen benutzen.
Kann weder mit einem noch mit beiden Würfeln ein Kästchen geschlossen werden, ist das Spiel beendet und der Spieler bekommt die Summe aller noch geöffneter Kästchen als Strafpunkte angekreidet.



Die Eingaben des Spielers werden ständig überprüft, so daß die Möglichkeit, sich selbst zu betrügen, von vornherein ausgeschlossen ist.
Am Ende eines relativ kurzen Durchganges kann sofort ein neues Spiel begonnen werden, ohne das Programm noch einmal neu zu starten.


```
10 REM********************
11 REM*   DECKEL DRAUF   *
12 REM*                  *
13 REM*  Copyright  by   *
14 REM*                  *
15 REM*   JAN KUENEN     *
16 REM*                  *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole *
19 REM*                  *
23 REM* Speicherbelegung *
24 REM*     5881 Bytes   *
25 REM********************
26 REM
100 CALL CLEAR
110 CALL SCREEN(15)
120 LN$="-------------------
----------"
130 A$="******* DECKEL DRAUF
 *******"
140 CALL CHAR(97,"0028003844
7C4444")
150 CALL CHAR(111,"0028007C4
444447C")
160 CALL CHAR(117,"002800444
4444438")
170 CALL CHAR(126,"FFFFFFFFF
FFFFFFF")
180 CALL CHAR(128,"808080808
080808")
190 CALL CHAR(129,"")
200 CALL CHAR(136,"")
210 CALL CHAR(137,"000000000
00000FF")
220 CALL CHAR(138,"FF")
230 CALL CHAR(139,"010101010
1010101")
240 CALL CHAR(140,"808080808
080808")
250 CALL CHAR(144,"183C7EFFF
F7E3C18")
260 CALL COLOR(2,5,1)
270 CALL COLOR(13,2,9)
280 CALL COLOR(14,2,13)
290 PRINT LN$:A$:LN$: :
300 PRINT "SPIELREGELN"
310 PRINT "-----------"
320 PRINT "ES WIRD MIT ZWEI
WuRFELN GE-SPIELT.": :
330 PRINT "WaHLEN SIE:": :
340 PRINT "A.EINEN DER BEIDE
N WERTE. ...B.DIE SUMME DER
WERTE.": :
350 PRINT "PROBIEREN SIE ALL
E ZAHLEN ..ZU LoSCHEN."
360 PRINT "WaRE EINE ZAHL NI
CHT MEHR ..MoGLICH,DANN IST
DAS SPIEL .ZUENDE UND SIE BE
KOMMEN SO-"
370 PRINT "VIELE STRAFPUNKTE
```

```
 ALS DIE ..SUMME DER RESTLIC
HEN ZAHLEN.": :
380 PRINT "<ENTER>=START."
390 CALL KEY(0,K,S)
400 IF K<>13 THEN 390 ELSE 2
330
410 CALL CLEAR
420 FOR I=1 TO LEN(A$)
430 CALL HCHAR(1,2+I,ASC(SEG
$(A$,I,1)))
440 NEXT I
450 CALL HCHAR(3,3,126,29)
460 CALL HCHAR(7,3,126,29)
470 CALL HCHAR(19,3,126,29)
480 CALL VCHAR(3,3,126,16)
490 CALL VCHAR(3,31,126,16)
500 FOR I=1 TO 3
510 CALL HCHAR(3+I,4,129,27)
520 NEXT I
530 FOR I=1 TO 11
540 CALL HCHAR(7+I,4,136,27)
550 NEXT I
560 FOR I=1 TO 8
570 CALL VCHAR(4,4+I*3,128,3
)
580 NEXT I
590 RETURN
600 CALL SOUND(250,262,1)
610 CALL SOUND(250,247,1)
620 CALL SOUND(75,262,1)
630 CALL SOUND(425,349,1)
640 CALL SOUND(250,262,1)
650 RETURN
660 FOR I=1 TO 9
670 A(I)=I
680 CALL HCHAR(5,2+3*I,48+I)
690 F=F+A(I)
700 NEXT I
710 RETURN
720 RANDOMIZE
730 K=0
740 D1=0
750 IF F<7 THEN 800
760 D1=INT(RND*6)+1
770 X=D1
780 K=5
790 GOSUB 860
800 D2=INT(RND*6)+1
810 D3=D1+D2
820 X=D2
830 K=19
840 GOSUB 860
850 RETURN
860 CALL HCHAR(9,K+2,137,7)
870 CALL HCHAR(17,K+2,138,7)
880 CALL VCHAR(10,K+1,139,7)
890 CALL VCHAR(10,K+9,140,7)
900 FOR I=1 TO 7
910 CALL HCHAR(9+I,K+2,32,7)
920 NEXT I
```

LISTINGS

```
930 ON X GOSUB 1700,1720,175
0,1790,1840,1900
940 RETURN
950 CALL SOUND(500,175,1)
960 CALL SOUND(600,147,1)
970 CALL SOUND(300,131,1)
980 RETURN
990 M=1
1000 FOR I=1 TO 9
1010 IF F>6 THEN 1070
1020 IF D2<>A(I)THEN 1160
1030 KC=D2
1040 M=5
1050 I=9
1060 GOTO 1160
1070 IF D1<>A(I)THEN 1100
1080 M=M+1
1090 KC=D1
1100 IF D2<>A(I)THEN 1130
1110 M=M+1
1120 KC=D2
1130 IF D3<>A(I)THEN 1160
1140 M=M+1
1150 KC=D3
1160 NEXT I
1170 ON M GOTO 1510,1190,118
0,1260,1220
1180 IF D1<>D2 THEN 1260
1190 A$="NUR DIE "&STR$(KC)&
" IST MoGLICH."
1200 GOSUB 1990
1210 GOTO 1500
1220 A$="SIE HABEN GLuCK MIT
 DER "&STR$(KC)
1230 GOSUB 600
1240 GOSUB 1990
1250 GOTO 1500
1260 A$="WELCHE ZAHL WaHLEN
SIE?"
1270 GOSUB 1990
1280 CALL KEY(0,KL,S)
1290 IF S=0 THEN 1280
1300 CALL SOUND(100,440,1)
1310 CALL HCHAR(22,1,32,32)
1320 KC=KL-48
1330 IF KC=D1 THEN 1420
1340 IF KC=D2 THEN 1420
1350 IF KC=D3 THEN 1420
1360 A$="DAS GEHT NICHT!"
1370 CALL SOUND(300,-2,1)
1380 GOSUB 1990
1390 FOR I=1 TO 300
1400 NEXT I
1410 GOTO 1260
1420 FOR I=1 TO 9
1430 L=0
1440 IF KC=A(I)THEN 1470
1450 L=1
1460 GOTO 1480
1470 I=9
```

```
1480 NEXT I
1490 IF L=1 THEN 1360
1500 A(KC)=0
1510 RETURN
1520 CALL HCHAR(22,1,32,32)
1530 CALL HCHAR(24,1,32,32)
1540 FOR J=1 TO 2
1550 FOR I=1 TO 9
1560 CALL HCHAR(8+I,J*14-8,1
36,9)
1570 NEXT I
1580 NEXT J
1590 RETURN
1600 GOSUB 1520
1610 F=0
1620 FOR I=1 TO 9
1630 IF A(I)=0 THEN 1660
1640 CALL HCHAR(5,2+I*3,A(I)
+48)
1650 GOTO 1670
1660 CALL HCHAR(5,2+I*3,129)
1670 F=F+A(I)
1680 NEXT I
1690 RETURN
1700 CALL HCHAR(13,K+5,144)
1710 RETURN
1720 CALL HCHAR(11,K+5,144)
1730 CALL HCHAR(15,K+5,144)
1740 RETURN
1750 FOR I=1 TO 3
1760 CALL HCHAR(9+I*2,K+1+I*
2,144)
1770 NEXT I
1780 RETURN
1790 FOR I=1 TO 2
1800 CALL HCHAR(7+I*4,K+3,14
4)
1810 CALL HCHAR(7+I*4,K+7,14
4)
1820 NEXT I
1830 RETURN
1840 FOR I=1 TO 2
1850 CALL HCHAR(7+I*4,K+3,14
4)
1860 CALL HCHAR(7+I*4,K+7,14
4)
1870 NEXT I
1880 CALL HCHAR(13,K+5,144)
1890 RETURN
1900 FOR I=1 TO 3
1910 CALL HCHAR(9+I*2,K+3,14
4)
1920 CALL HCHAR(9+I*2,K+7,14
4)
1930 NEXT I
1940 RETURN
1950 A$="KEINE WAHL MoGLICH!
"
1960 GOSUB 950
1970 GOSUB 1990
1980 RETURN
1990 CALL HCHAR(22,1,32,32)
2000 FOR I=1 TO LEN(A$)
2010 CALL HCHAR(22,2+I,ASC(S
EG$(A$,I,1)))
2020 NEXT I
2030 FOR I=1 TO 200
2040 NEXT I
2050 RETURN
2060 IF F=0 THEN 2140
2070 B$="ES IST VORBEI!"
2080 A$="SIE BEKOMMEN "&STR$
(F)&" STRAFPUNKTE."
2090 IF F<>1 THEN 2110
2100 A$="SIE HABEN EINEN STR
AFPUNKT."
2110 GOSUB 1990
2120 GOSUB 2260
2130 GOTO 2170
2140 A$="SIE HABEN ES GESCHA
FFT!"
2150 GOSUB 600
2160 GOSUB 1990
2170 B$="NOCH EIN SPIEL? <J-
N> "
2180 GOSUB 2260
2190 CALL KEY(0,KJ,S)
2200 IF S=0 THEN 2190
2210 IF KJ<>74 THEN 2240
2220 GOSUB 1520
2230 RETURN
2240 CALL CLEAR
2250 END
2260 CALL HCHAR(24,1,32,32)
2270 FOR I=1 TO LEN(B$)
2280 CALL HCHAR(24,2+I,ASC(S
EG$(B$,I,1)))
2290 NEXT I
2300 FOR I=1 TO 200
2310 NEXT I
2320 RETURN
2330 GOSUB 410
2340 GOSUB 660
2350 GOSUB 720
2360 GOSUB 990
2370 IF M=1 THEN 2420
2380 GOSUB 1600
2390 IF F=0 THEN 2400 ELSE 2
350
2400 GOSUB 2060
2410 GOTO 2340
2420 GOSUB 1950
2430 GOTO 2400
```

FREEWARE: QUELLDATEIEN FÜR DIE GRAM-KARTE

Als Freeware sind die Quelldateien zweier sehr nützlichen Programme für den TI-99 mit 128/512k-GRAM-Karte herausgekommen. Um sie vorzustellen, muß man zuerst etwas über die GRAM-Karten sagen. Diese Karte gibt einem die Möglichkeit, ohne weitere Hardware Module z.B. selbst zu schreiben oder umzuändern. Man kann sie als Masterplatine für fast alle bisher für den TI erhältlichen Module betrachten. Weiterhin besteht mit der GRAM-Karte die Möglichkeit, das ROM „umzuändern", d.h. man kopiert es einfach in die GRAM-Karte. Der Computer verwendet dann die Kopie anstatt seine eigenen Roms. Dies gibt einem die Möglichkeit, das Rom und auf gleiche Weise auch die GROMs nach dem eigenen Bedarf zu verändern.

Wie wahrscheinlich schon bekannt, werden die GROMs in der Zwischensprache GPL programmiert. Besitzt man also einen GPL-Assembler, so kann man seine eigenen GROM-Module programmieren.
Eines solcher in GPL geschriebener Module ist der Converter.
Seine Quelldatei erhält man ebenfalls auf der Freeware-Diskette. Dieses Modul hat vor allem für DFÜ-Betreiber große Vorteile. Es wandelt List-Files von BASIC-Programmen, solche, wie man sie durch ein Telefon-Modem erhält, in das PROGRAM-Format um. Man erhält dadurch ein wieder lauffähiges Programm. Eventuell auftretende Fehler werden anhand einer REM-Zeile in dem neu entstandenen BASIC-Programm angegeben. für den Konvertiervorgang hat man drei BASIC-Dialekte zur Auswahl. TI-BASIC, EX-BASIC und M-BASIC. Unter M-BASIC versteht man Microsoft-BASIC, welche geringfügige Unterschiede zum EX-BASIC aufweist. Dieses Modul erspart dem DFÜ-Benutzer somit das mühselige Abtippen der Programme, die er durch seinen Akustik-Koppler erhält. Die einzige Arbeit besteht danach noch darin, eventuelle Fehler auszubessern. Dies artet jedoch nicht in Schwerstarbeit aus, da diese Fehler klar und deutlich im Listing zu erkennen sind. Ein weiterer Vorteil des Konverters ist, daß man BASIC-Programme anhand des Bildschirmeditors des Editor Assemblers oder des TI-Writers erstellen kann. Durch diesen Editor kann man das so entstandene Programm einfacher editieren und ausbessern. Der Nachteil dabei ist, daß der Comand-Modus entfällt und man das Programm, bevor man es testen will, zuerst konvertieren muß.
Die wohl interessanteste Möglichkeit, wie man die GRAM-Karte verwenden kann, ist eindeutig die GRAM-Disk. Sie ist, ähnlich der RAM-Disk, voll als Disklaufwerk anzusprechen und zu verwenden. Der Unterschied dabei liegt jedoch im Bereich, den die GRAM-Disk verwendet. Für die Speicherung von Files stehen 31k zur Verfügung. Diese setzen sich aus dem GROM-Bereichen von G>6000–G>DFFF zusammen. Die oberste Bank von G>E000–G>FFFF ist für das EX-BASIC reserviert, da dieses, wenn es geladen ist bzw. wird, diese Bank benötigt. Abzüglich der 32k, die der Bereich von G>6000–G>DFFF umfaßt, ist noch die GRAM-Disk-Software, die, ohne sie mit dem GPL-Assembler geändert zu haben, auf die GROM-Adresse G>9820 geladen werden muß. Die Bezeichnung GROM ist insofern nicht mehr ganz richtig, da die GROMs auf der GRAM-Karte ebenfalls aus GRAMs bestehen, die als GROMs codiert sind. Der wohl spektakulärste Vorteil der GRAM-Disk besteht darin, daß man aus BASIC den Speicherplatz eines Modules zur Datenspeicherung verwenden kann. Diese Ausnützung eines bisher brachliegenden Speicherplatzes ist nur mit der GRAM-Disk möglich, da diese, wie oben schon erwähnt, keine GROMs beinhaltet, sondern nur beschreibbare GRAMs. Wer eine RAM-Disk kennt und auch benützt, wird sehr schnell auch die GRAM-Disk zu schätzen wissen. Diese steht, wie die RAM-Disk, in keinem Geschwindigkeitsverhältnis zur normalen Diskettenstation. Dadurch, daß die GRAM-Disk ebenfalls nur im Speicher operiert und so kein Gerät ansteuern braucht, wird ein traumhafter Geschwindigkeitsgewinn erreicht. Die GRAM-Disk ist bis zu 5mal schneller als das normale Diskettenlaufwerk. Einer Verwendung als weiteres Laufwerk steht daher nichts im Wege. Die Verwender der 128k-GRAM-Karte haben dabei jedoch eine Einschränkung. Dadurch, daß die GRAM-Disk nur einen File beinhalten kann und die 128k-GRAM-Karte nur Banks für zwei Module zur Verfügung hat, kann man, parallel zu einem weiteren Modul, nur eine GRAM-Disk verwenden. Das bedeutet, daß man die GRAM-Disk nur zur Speicherung eines Files verwenden kann. Einen Vorteil haben die Benützer mehrerer GRAM-Karten bzw. einer 512k-GRAM-Karte. Diese haben die Möglichkeit, nach entsprechenden Abänderungen der Software, mehrere GRAM-Disks in die GRAM-Karte zu laden und auch zu verwenden. *Oliver Huber*

TRAUMHAFT SCHNELL

Bezugsquelle:
Elektronik-Service
Peter Kleinschmidt
Linning 37
D-4044 Kaarst 2

MAGISCHES QUADRAT

Wenn man „Magisches Quadrat" hört, denkt man sofort an die Zahlenfelder, die in jeder Reihe die gleiche Summe ergeben.
Doch dieses TI-Basic-Spiel hat damit nur wenig gemeinsam. Es ist viel schwieriger . . .
Nach dem Programmstart bauen sich auf dem Bildschirm zwei Quadrate auf. Ein kleines mit einem vorgegebenen Farbmuster und ein großes Quadrat.
Wenn Sie nun eine Taste zwischen 1 – 9 drücken, ändern sich nach einem bestimmten Schema die farbigen Teile des größeren Quadrates.
Hier beginnt nun Ihre Aufgabe:
Schauen Sie sich das Muster des kleinen Quadrates genau an und versuchen Sie, es im Spielfeld abzubilden. Dies gelingt aber nur, wenn Sie das Schema der Farbänderungen richtig erkannt haben oder lange genug vor lauter Verzweiflung in die Tasten gehauen haben, bis der Zufall Ihnen zu Hilfe kommt. Doch darauf würde ich mich nicht verlassen. Auch im Lotto hoffen jede Woche viele Tipper auf den Zufallstreffer und nur wenige werden dabei wirklich reich.




```
1Ø REM*********************
11 REM*  DAS VERFLIXTE    *
12 REM*   FARBQUADRAT     *
13 REM*  Copyright  by    *
14 REM* Burkhardt Knedel  *
15 REM*                   *
17 REM*Benoetigte Geraete*
18 REM* TI99/4A  Konsole  *
19 REM*                   *
23 REM* Speicherbelegung *
24 REM*     2879 Bytes    *
25 REM*********************
26 REM
1ØØ CALL SOUND(12ØØ,11Ø,1,33
Ø,1,55Ø,1)
11Ø CALL SCREEN(16)
12Ø CALL CLEAR
13Ø PRINT "       MAGISCHES Q
UADRAT": : :"SIE MUESSEN DIE
 FARBEN DES": :"KLEINEN MUST
ERS OBEN RECHTS": :
14Ø PRINT "NACHVOLLZIEHEN.":
 : :"DIE TASTEN 1-9 AENDERN
NACH": :"EINEM BESTIMMTEN SC
HEMA DIE": :"FARBEN.": : :
15Ø PRINT "ERKENNEN SIE ES!!
!": : : :
16Ø PRINT "BITTE EINE TASTE
DRUECKEN."
17Ø CALL KEY(Ø,X,S)
18Ø IF S=Ø THEN 17Ø
19Ø FOR I=6 TO 12
2ØØ CALL SOUND(3ØØ,I*11Ø,1)
21Ø NEXT I
22Ø CALL VCHAR(1,1,32,768)
23Ø CALL CHAR(39,"FFFFFFFFFF
FFFFFF")
24Ø RANDOMIZE
25Ø DIM A(9)
26Ø FOR I=1 TO 9
27Ø IF RND>.5 THEN 3ØØ
28Ø A(I)=1
29Ø GOTO 31Ø
3ØØ A(I)=3
31Ø NEXT I
32Ø CALL SCREEN(12)
33Ø CALL COLOR(2,7,7)
34Ø CALL COLOR(12,5,5)
35Ø A$="MAGISCHES QUADRAT"
36Ø FOR I=1 TO LEN(A$)
37Ø CALL HCHAR(1,7+I,ASC(SEG
$(A$,I,1)))
38Ø NEXT I
39Ø A$="WELCHE TASTE?"
4ØØ FOR I=1 TO LEN(A$)
41Ø CALL HCHAR(24,5+I,ASC(SE
G$(A$,I,1)))
42Ø NEXT I
```

LISTING

```
430 CALL HCHAR(24,19,61)
44Ø A$="VERSUCH NR="
45Ø FOR I=1 TO LEN(A$)
46Ø CALL HCHAR(21,5+I,ASC(SE
G$(A$,I,1)))
47Ø NEXT I
48Ø CALL HCHAR(4,27,4Ø,3)
49Ø CALL HCHAR(5,27,4Ø)
5ØØ CALL HCHAR(5,29,4Ø)
51Ø CALL HCHAR(6,27,4Ø,3)
52Ø CALL HCHAR(5,28,12Ø)
53Ø CALL HCHAR(3,8,39,17)
54Ø CALL VCHAR(4,24,39,15)
55Ø CALL HCHAR(19,8,39,17)
56Ø CALL VCHAR(4,8,39,15)
57Ø GOTO 12ØØ
58Ø V$=STR$(V)
59Ø FOR I=1 TO LEN(V$)
6ØØ CALL HCHAR(21,16+I,ASC(S
EG$(V$,I,1)))
61Ø NEXT I
62Ø CALL KEY(Ø,X,S)
63Ø IF S=Ø THEN 62Ø
64Ø IF (X<49)+(X>57)=-1 THEN
 62Ø
65Ø CALL HCHAR(24,2Ø,X)
66Ø CALL SOUND(-4Ø,44Ø,1)
67Ø ON X-48 GOTO 68Ø,75Ø,79Ø
,86Ø,9ØØ,98Ø,1Ø2Ø,1Ø9Ø,113Ø
68Ø FOR I=1 TO 2
69Ø GOSUB 144Ø
7ØØ NEXT I
71Ø FOR I=4 TO 5
72Ø GOSUB 144Ø
73Ø NEXT I
74Ø GOTO 12ØØ
75Ø FOR I=1 TO 3
76Ø GOSUB 144Ø
77Ø NEXT I
78Ø GOTO 12ØØ
79Ø FOR I=2 TO 3
8ØØ GOSUB 144Ø
81Ø NEXT I
82Ø FOR I=5 TO 6
83Ø GOSUB 144Ø
84Ø NEXT I
85Ø GOTO 12ØØ
86Ø FOR I=1 TO 7 STEP 3
87Ø GOSUB 144Ø
88Ø NEXT I
89Ø GOTO 12ØØ
9ØØ I=2
91Ø GOSUB 144Ø
92Ø FOR I=4 TO 6
93Ø GOSUB 144Ø
94Ø NEXT I
95Ø I=8
96Ø GOSUB 144Ø
97Ø GOTO 12ØØ
98Ø FOR I=3 TO 9 STEP 3
99Ø GOSUB 144Ø
1ØØØ NEXT I
1Ø1Ø GOTO 12ØØ
1Ø2Ø FOR I=4 TO 5
1Ø3Ø GOSUB 144Ø
1Ø4Ø NEXT I
1Ø5Ø FOR I=7 TO 8
1Ø6Ø GOSUB 144Ø
1Ø7Ø NEXT I
1Ø8Ø GOTO 12ØØ
1Ø9Ø FOR I=7 TO 9
11ØØ GOSUB 144Ø
111Ø NEXT I
112Ø GOTO 12ØØ
113Ø FOR I=5 TO 6
114Ø GOSUB 144Ø
115Ø NEXT I
116Ø FOR I=8 TO 9
117Ø GOSUB 144Ø
118Ø NEXT I
119Ø GOTO 12ØØ
12ØØ I=Ø
121Ø FOR Z=4 TO 14 STEP 5
122Ø FOR S=9 TO 19 STEP 5
123Ø I=I+1
124Ø CALL SOUND(2ØØ,44Ø,1Ø)
125Ø FOR Y=Ø TO 4
126Ø CALL HCHAR(Z+Y,S,A(I)*4
Ø,5)
127Ø NEXT Y
128Ø NEXT S
129Ø NEXT Z
13ØØ CALL HCHAR(24,2Ø,32)
131Ø IF (A(1)=1)+(A(2)=1)+(A
(3)=1)+(A(4)=1)+(A(5)=3)+(A(
6)=1)+(A(7)=1)+(A(8)=1)+(A(9
)=1)=-9 THEN 135Ø
132Ø V=V+1
133Ø CALL SOUND(5ØØ,888,1)
134Ø GOTO 58Ø
135Ø CALL SOUND(15ØØ,-6,1Ø)
136Ø FOR I=1 TO 1Ø
137Ø CALL SCREEN(2)
138Ø CALL SOUND(3ØØ,I*2ØØ,Ø,
I*2ØØ+5Ø,Ø,I*2ØØ+8Ø,Ø)
139Ø CALL SCREEN(12)
14ØØ NEXT I
141Ø CALL HCHAR(1,1,32,768)
142Ø PRINT "HERZLICHEN GLUEC
KWUNSCH!": : :"SIE HABEN ES
MIT";V;: : :"VERSUCHEN GESCH
AFFT.": : : : : : :
143Ø END
144Ø IF A(I)=1 THEN 147Ø
145Ø A(I)=1
146Ø RETURN
147Ø A(I)=3
148Ø RETURN
```

AUSGABE-BAUSTEIN FÜR DEN TI/99/4A

Mit diesem Ausgabemodul ist es möglich, z.B. Roboter zu steuern. Es werden lediglich die Konsole und ein Sichtgerät (Fernseher oder Monitor) benötigt.

Der oder die lichtempfindlichen Widerstände (LDR-03) werden an bestimmten Punkten am Bildschirm befestigt (z.B. mit Tesafilm oder Fischertechnikbausteinen).
Diese Punkte werden dann vom Basicprogramm angesteuert. Je nach Schaltung des Relais, bzw. je nach Programm, kann mit hellen oder dunklen Feldern auf dem Bildschirm das Relais zum Schalten gebracht werden.
Bei dem Ablauf eines Programms muß darauf geachtet werden, daß der Bildschirm nicht im vollen Sonnenlicht steht, da es sonst zu unkontrollierten Schalten des Relais kommt.
Die gesamte Steuerschaltung kann mit Fischertechnikbauelementen aufgebaut werden. Es wird dazu nur der Elektronikkasten bzw. Grundkasten gebraucht.
Es kann aber auch eine Platine erstellt werden, die mit den einzelnen Bauelementen bestückt wird. Die Bauelemente gibt es z.B. bei Konrad oder Völkner.
Beim Aussuchen des Relais sollte darauf geachtet werden, daß der Kollektorstrom des Transistors nicht zu groß wird (nicht über 0.1 A). Falls dies doch der Fall sein sollte, kann man auf größere Transistoren ausweichen, z.B. BD 135.
Das Potentiometer der Schaltung dient zum Abgleichen auf die Helligkeit des Monitors bzw. Fernsehers.
Man kann die Schaltung beliebig auf mehrere Ausgänge erweitern.

RÜCKMELDUNG INSTALLIEREN

Falls man eine Robotersteuerung realisieren möchte, muß man zusätzlich eine Rückmeldung installieren.
Diese Rückmeldung kann man sehr gut über den Joystickport führen.
Den dafür benötigten Trapezstecker bezieht man auch bei Konrad oder Völkner.
Die im Schaltplan angegebenen Nummern befinden sich auf der Rückseite des Steckers.
Über das Basicprogramm wird mit Call Joyst der Port abgefragt.

Rainer Pahlmann

Stückliste für 1-Kanalausgabe

1 BC 107 (Transistor)
1 1 N 4007 (Diode)
1 LDR 03 (Photowiderstand
1 5K Potentiometer (regelbarer Widerstand)
1 2.2K Widerstand
1 8.2K Widerstand
1 beliebiges 12V (besser 6 V) Relais

```
10 ! *********************
11 ! *                   *
12 ! * ALLGEM. STEUERPRG. *
13 ! *                   *
14 ! *    COPYRIGHT BY   *
15 ! *                   *
16 ! *  RAINER PAHLMANN  *
17 ! *                   *
18 ! *********************
100 MA=9
110 FOR I=0 TO 14 :: CALL CO
LOR(I,16,2):: NEXT I :: CALL
 SCREEN(2)
120 CALL CLEAR
130 CALL CHAR(65,"FFFFFFFFFF
FFFFFF")
140 FOR Z=1 TO MA
150 READ A$
160 CALL CLEAR
170 FOR I=1 TO 4
180 A=VAL(SEG$(A$,I,1))
190 IF A=0 THEN 210
200 ON I GOSUB 250,280,310,3
40
210 NEXT I
220 CALL KEY(1,K,S):: IF K<>
18 THEN 220 ! WENN K=18,DANN
 NAECHSTE PUNKTANSTEUERUNG
230 NEXT Z
240 RESTORE :: GOTO 140
250 CALL HCHAR(22,1,65,3)
260 CALL HCHAR(32,1,65,3)
270 RETURN
280 CALL HCHAR(22,8,65,3)
290 CALL HCHAR(23,8,65,3)
300 RETURN
310 CALL HCHAR(22,14,65,3)
320 CALL HCHAR(23,14,65,3)
330 RETURN
340 CALL HCHAR(22,22,65,3)
350 CALL HCHAR(23,22,65,3)
360 RETURN
370 ! BEISPIELDATEN
380 ! DATA 0000,1000,0100,00
10,0001
390 DATA 0000,0110,0001,1000
,1101,1010,1001,0003,0101
```

Rückmeldung über Joystickport

Abfrage:
CALL JOYST(2,X,Y)
CALL KEY(2,K,S)

:Pin 2

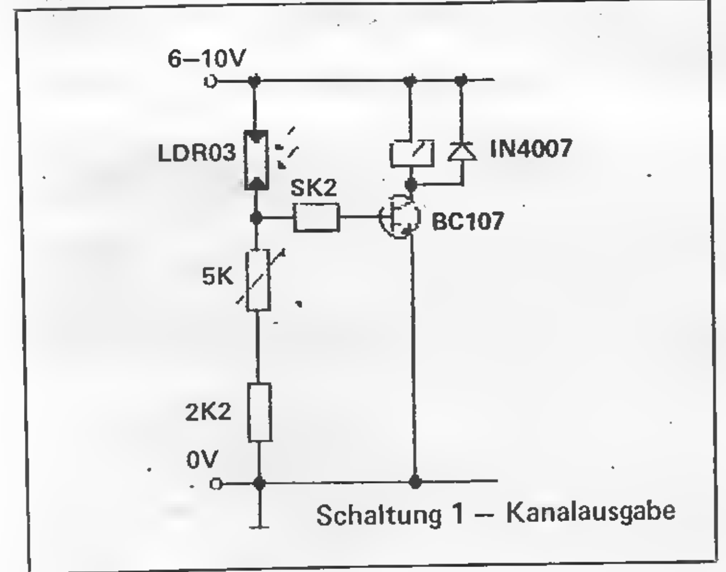
Pin 3:X=+4
Pin 4:K=18
Pin 9:Y=+4

CALL JOYST(1,X,Y)
CALL KEY(1,K,S)

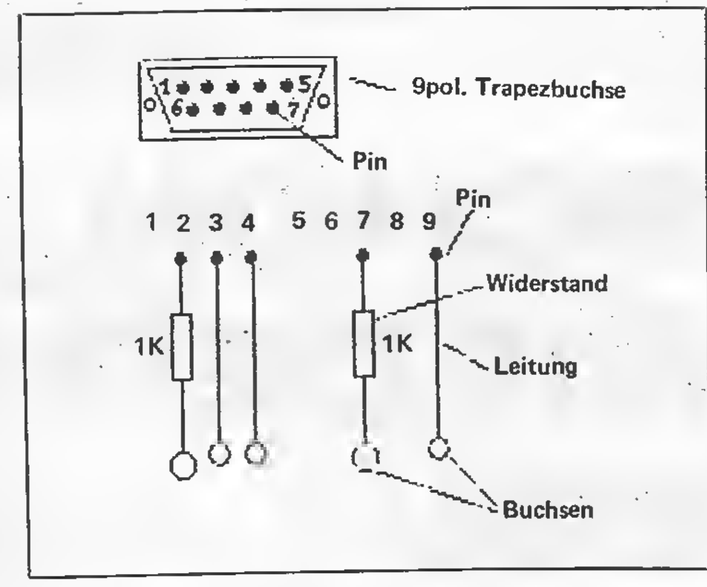
:Pin 7

Pin 3:X=+4
Pin 4:K=18
Pin 9:Y=+4

Z.B.: K=18, wenn Pin 2 mit Pin 4 verbunden ist.



Schaltung 1 – Kanalausgabe




```
10 ! ********************
11 ! *                  *
12 ! *SPEZIELLES STEUERPRG*
13 ! *                  *
14 ! *    COPYRIGHT BY   *
15 ! *                  *
16 ! *  RAINER PAHLMANN  *
17 ! *                  *
18 ! ********************
100 FOR I=Ø TO 14 :: CALL CO
LOR(I,16,2):: NEXT I :: CALL
 SCREEN(2)
110 CALL CLEAR
120 CALL CHAR(98,"FFFFFFFFFF
FFFFFF"
130 DISPLAY AT(1Ø,1):"MAGNET
...AUS":"POLWENDE AUS":"DREH
UNG  AUS":"ARM......AUS"
140 CALL KEY(Ø,K,S):: IF S=Ø
 THEN 14Ø
150 DISPLAY AT(1Ø,1):"MAGNET
...AUS":"POLWENDE EIN":"DREH
UNG..EIN":"ARM......AUS"
160 CH=98 :: GOSUB 38Ø :: GO
SUB 41Ø
170 CALL KEY(2,K,S):: IF S=Ø
 THEN 17Ø :: IF K<>18 THEN 1
7Ø
180 DISPLAY AT(1Ø,1):"MAGNET
...AUS":"POLWENDE AUS":"DREH
UNG..AUS":"ARM......EIN"
190 CH=32 :: GOSUB 38Ø :: GO
SUB 41Ø :: CH=98 :: GOSUB 44
Ø
200 CALL JOYST(1,X,Y):: IF X
+Y=Ø THEN 2ØØ :: IF X<>4 THE
N 2ØØ
210 DISPLAY AT(1Ø,1):"MAGNET
...EIN":"POLWENDE EIN":"DREH
UNG..AUS":"ARM......EIN"
220 CH=98 :: GOSUB 35Ø :: GO
SUB 38Ø
230 CALL JOYST(1,X,Y):: IF X
+Y=Ø THEN 23Ø :: IF Y<>4 THE
N 23Ø
240 DISPLAY AT(1Ø,1):"MAGNET
...EIN":"POLWENDE AUS":"DREH
UNG..EIN":"ARM......AUS"
250 CH=32 :: GOSUB 38Ø :: GO
SUB 44Ø :: CH=98 :: GOSUB 41
Ø
260 CALL KEY(1,K,S):: IF S=Ø
 THEN 26Ø :: IF K<>18 THEN 2
6Ø
270 DISPLAY AT(1Ø,1):"MAGNET
...EIN":"POLWENDE AUS":"DREH
UNG..AUS":"ARM......EIN"
280 CH=32 :: GOSUB 41Ø :: CH
=98 :: GOSUB 44Ø
290 CALL JOYST(1,X,Y):: IF X
+Y=Ø THEN 29Ø :: IF X<>4 THE
N 29Ø
300 DISPLAY AT(1Ø,1):"MAGNET
...AUS":"POLWENDE EIN":"DREH
UNG..AUS":"ARM......EIN"
310 CH=32 :: GOSUB 35Ø :: CH
=98 :: GOSUB 38Ø
320 CALL JOYST(1,X,Y):: IF X
+Y=Ø THEN 32Ø :: IF Y<>4 THE
N 32Ø
330 DISPLAY AT(1Ø,1):"MAGNET
...AUS":"POLWENDE AUS":"DREH
UNG..AUS":"ARM......AUS"
340 FOR I=2Ø TO 24 :: CALL H
CHAR(I,1,32,32):: NEXT I ::
FOR I=1 TO 5ØØ :: NEXT I ::
GOTO 14Ø
350 CALL HCHAR(22,1,CH,3)
360 CALL HCHAR(23,1,CH,3)
370 RETURN
380 CALL HCHAR(22,8,CH,3)
390 CALL HCHAR(23,8,CH,3)
400 RETURN
410 CALL HCHAR(22,14,CH,3)
420 CALL HCHAR(23,14,CH,3)
430 RETURN
440 CALL HCHAR(22,22,CH,3)
450 CALL HCHAR(22,23,CH,3)
460 RETURN
```

SCHON WIEDER EIN COMPILER

Dieser Compiler ist dafür gedacht, Programme (die in C geschrieben wurden) in Assembler Code zu wandeln. Danach wird der Assembler Code nur noch assembliert und man hat ein Maschinencode-Programm. Welche Möglichkeiten man in dieser, für den TI-99 neuen, Sprache hat, möchte ich hier aufzeigen.

Der C-Compiler erschien in der ersten Version 1985 auf dem Freeware-Markt. Dieses Jahr nun folgte eine zweite Version, die auch, neben der verbesserten Compilierung, Lauffähigkeit auf den sog. Editor/Assembler-Simulatoren(Beaxs und Funlwriter) garantiert.
Auf den Disketten ist eine sehr ausführliche Anleitung enthalten, die alle Programme und Hilfsroutinen erklärt und die Besonderheiten der C99-Version zeigt.
Natürlich ist es schwierig, ohne sonstige Vorbildung von C etwas Vernünftiges auf die Beine zu stellen, deshalb empfiehlt es sich, neben der Anleitung auch noch C-Literatur durchzuarbeiten.
Weiterhin sollte man sich einen strukturierten Programmierstil zulegen, da C eine hochstrukturierte Sprache ist und auch so behandelt werden möchte (ansonsten bedankt sich der Compiler mit einigen sehr ausführlichen Fehlermeldungen . . .).
C wird hauptsächlich auf größeren Rechnern eingesetzt (wie z.B. Rechner mit CP/M, UNIX, u.a. als Betriebssystem).
Die Sprache selbst hat eine schon relativ lange Geschichte, die bis ins Jahr 1972 zurückreicht.
C99 entstand aus einer small-c Version. Natürlich kann dieser Compiler nicht soviel bieten wie Compiler für Großrechner, aber was diese „kleine“ Compiler-Version leistet, ist doch schon recht beachtlich!
Für diejenigen, die C nicht kennen, sei folgendes gesagt:
C wird in verschiedene Funktionen gegliedert, jede Funktion erfüllt einen abgeschlossenen Programmteil. Werden diese Teile nun in bestimmten Reihenfolgen aufgerufen, so entsteht ein komplexes Programm.
Bei den Funktionen liegt der Vergleich zum Extended BASIC nahe, indem der Befehl
CALL UPRO() und
SUB UPRO()
den selben Zweck erfüllen!
PASCAL wird oft mit C verglichen – von der Programmstruktur mag dies ja stimmen, da aber PASCAL „eingebaute“ Funktionen zur Verfügung stellt und C nicht, trennen sich hier die Wege.

ACHTUNG: SEMIKOLON!

Die Unterschiede zu BASIC liegen in einem Punkt darin, daß die Variablen einen bestimmten DATENTYP zugeordnet werden müssen. Dabei gibt es in C folgende Typen:

int (für Integerzahlen)
float (für Fließkommazahlen)
char (für Zeichen und Strings)

Im Gegensatz zu C gibt es in C99 kein float, aber dieser Nachteil wird durch ein Programm, das Floating Point Routinen ermöglicht, wieder ausgeglichen. Um nochmals auf die Datentypen zurückzukommen: Die Variablen müssen also zuerst typisiert und dann erst definiert werden!

```
Z.B.:  int x,y,z;
       char buff(81);
       char *name;
```

Vielleicht ist euch aufgefallen, daß hinter jedem Befehl ein Semikolon(;) steht. Dieses sagt dem Compiler, daß hier ein Befehl abgeschlossen ist – aber Vorsicht: Nicht überall folgt ein Strichpunkt und das ist ärgerlich beim Compilieren . . . obwohl dies bei kurzen Programmen relativ schnell geschieht!
Der Stern vor dem Variablenname bedeutet, daß die Variable Zeiger auf etwas ist.
Gehen wir nun dazu über, einige Funktionen näher kennenzulernen und somit die Möglichkeiten von C99 zu zeigen.
Die vordefinierten Funktionen erleichtern den Zugriff aus C auf:

- Bildschirm
- I/O Operationen
- Grafik
- Sprache
- Sound

Wenn ein C99-Programm gestartet wird, ist der 40-Zeichen-Modus eingeschaltet. Folgende Funktionen erleichtern den Zugriff:

- getchar();
  wartet auf Tastendruck und gibt ASCII Code zurück
- putchar(c);
  schreibt ASCII Code auf den Bildschirm
- gets(buff);
  liest einen String in den Buffer
- puts(string);
  gibt einen String auf dem Schirm aus
- locate(r,c);
  plaziert den Cursor an die angegebene Stelle
- poll(c);
  wartet auf Tastendruck und übergibt Tastenwert
- tscrn(f;b);
  ändert die Farbe im Textmode auf f(Vordergrund) und b(Hintergrund).

Folgende I/O-Operationen erleichtern auch die Übergabe von Daten auf andere Peripheriegeräte:

- unit=fopen(name, mode);
  eröffnet eine Datei im speziellen Mode (z.B. eine Disk-Datei im DISPLAY/VARIABLE 80 Mode)
- c=fclose(unit);
  schließt eine Datei
- c=getc(unit);
  liest einen Charakter von einer geöffneten Datei
- c=putc(c,unit);
  schreibt einen Charakter auf eine Datei
- c=fgets(buff,col,unit);
  liest einen String vom

File in den Buffer – col gibt dabei die Anzahl der Charakter –1 an
- c=fputs(string, unit); schreibt einen String auf eine Datei
- fread(buff,len,unit); liest den nächsten String (Anzahl der Charakter:len) in den Buffer
- c=fwrite(buff,len,unit); wie fread, nur umgekehrt
- fseek(unit,recnr); nichts anderes als Restore in C!
- fdelete(filename); löscht ein File auf der Diskette
- c=feof(unit); testet, ob eine Datei schon am Ende angelangt ist
- c=ferrc(unit); gibt den Fehlercode zurück, falls ein Fehler auftritt
- rewind(unit); stellt den fseek Zeiger auf das erste Byte einer Datei ein.

Dies soll mal als erster Überblick genügen, denn gerade die C99-Version 2 bietet noch sehr viel mehr an Funktionen, die den Rahmen sprengen würden. Allein mit diesen Funktionen wäre es denkbar, ein (z.B.) Textverarbeitungssystem zu erstellen – und dies, wie man zugeben muß, relativ einfach, sehr effektiv und schnell.
Aber überall wo Licht ist, gibt es auch Schatten – kommen wir nun auch hierzu.
Der C99-Compiler unterstützt noch nicht:
- mehr als eine Dimension in einem Feld
- kein float als Standardfunktion (zuladbar!)
- bei Funktionsaufrufen werden nur Werte vom Typ int zurückgegeben
- die Operatoren && , ?: , :: kennt der Compiler nicht
- Zuweisungsoperatoren werden nicht erkannt
- Speicherklassen (extern, static, auto, . . .)
- Goto Statement

und andere, kleinere Abweichungen gegenüber Standard C.

Diese Aufzählung sieht jetzt zwar relativ lang aus, aber man kommt auch gut ohne die nicht vorhandenen Operatoren und Befehle aus, zumal man diese auch noch im Nachhinein als Funktion definieren kann.
Hier weicht C99 wieder vom Standard ab und bietet eine Möglichkeit an, Assembler und C zu mischen!
Dies erfolgt durch die C-unübliche Befehlsfolge:

```
#asm
  BLWP@VMBW(Beispiel)
#endasm
```

Viele Beispielprogramme in C sind auf den Disketten enthalten, die die neue Sprache näherbringen sollen.

NUR AM ANFANG KOMPLIZIERT

Dazu zählt auch ein Benchmark Test mit 10 Iterationen. Es liegt ein Vergleich nahe zwischen dem immer beliebter werdenden FORTH und dem C99. Clint Pulley, der Autor des Programmes, wagte den Vergleich und es ergab sich, daß C99 doppelt so schnell ist wie FORTH!
Abschließend kann man behaupten, daß dieses C-Paket voll und ganz abgerundet ist – sowohl in der Software als auch von dem sehr informativen Begleitmaterial (englisch). Man kann mit ruhigem Gewissen diesen Compiler weiterempfehlen, denn so einfach wird der Einstieg in C (und somit auch in Assembler) nicht mehr angeboten.
Natürlich, ich habe es am Anfang schon erwähnt, wird ein strukturiertes Programmieren und evtl. Kenntnisse von BASIC und C (denn darauf baut es fast auf) verlangt.
Am Anfang ist alles etwas kompliziert, aber die Durchsicht kommt schnell!
Clint Pulley hat somit ein Programm geschaffen, das eine große Bereicherung des Softwaremarktes vom TI-99 darstellt. *Oliver Siffrin*

NACHTRAG ZUM M-LADER

Um beim M-Lader eine bessere Kompatibilität zum Disk-Controller zu erreichen, stellte es sich als zweckmäßig heraus, den Segmentaufbau im VDP-RAM um ein '0-Byte' am Anfang zu ergänzen. Dadurch wird erreicht, daß u.a. der Buffer des Diskcontrollers nicht mehr der mit der niedrigsten Adresse, sein muß.

Demzufolge haben sich folgende fünf Zeilen des M-Laders verändert:
Zusätzlich wird die Zeile 26855 benötigt:
26855 DISK=VSEG=8192
Folgende Zeilen wurden modifiziert:
26900 CALL PEEKV (C,VSEG, AL,AR,A1, A2,E1,E2)
26910 IF DISK* (VSEG=170) THEN 26770
Die Zeilen 26920 und 26930 entfallen vollständig. Durch diese Anpassung haben sich beim 'COPY'-Befehl folgende Änderungen ergeben:
COPY
fügt Anweisungen ein, die außerhalb des M-Laders stehen
Format: COPY,BS.
[dezimale VDP Segment Nummer, >0]
bzw.: COPY,[Filename (DATA-Zeilen in LIST-Format)]
Die Anweisung kann 9-fach geschachtelt sein. Die momentane Ebene der Schachtelung wird in Zeile 26600 sowie bei Fehlermeldungen als FILE-Nummer gezeigt.
Verwendung zum Einlesen aus externen Programmteilen, z.B. Standard-Definitionen über "EQU" für etwa "STRASG" u.ä.
Die externen DATA-Zeilen können in einem File in LIST-Format gespeichert sein. Dabei darf die Zeilenlänge ca. 65 Zeichen nicht überschreiten und die letzte Zeile kein Komma enthalten, sollte also z.B. "30000 REM" lauten.
Die externen DATA-Zeilen können auch im VDP in Programm-Format als BASIC-Segment gespeichert sein. Letzteres erreichen Sie unter BASIC folgendermaßen:

- Sie benutzen ein Standardverfahren, um ein BASIC-Programm im Speicher zu erhalten, das in DATA-Zeilen die M-Lader-Anweisungen enthält
- Sie geben folgende Befehle ein
  CALL PEEK(−31888, A,S)
  CALL PEEK (−31952, Q,W,E,R)
  ENDE=256*Q+W−8
  END1=INT(ENDE/256)
  END2=ENDE−256 *END1
  CALL LOAD (−31888, END1,END2)
  CALL POKEV(ENDE +1,O,A,S,Q,W,E,R)
  NEW
- Damit ist ein BASIC-Programm gespeichert und der jetzt reduzierte VDP-Speicher kann ganz normal das nächste Programm aufnehmen, z.B. den M-Lader durch "OLD CS1".

Jedes so gespeicherte Programm belegt ein BASIC-Segment (BS), durchnumeriert vom neuesten zum ältesten. Beachten Sie, daß bei Verwendung von Disketten-Laufwerken deren I/O-Buffer stets auch ein Segment belegt, z.B. "BS.2" im obigen Diagramm.

VDP-Speicher	
BS.2:	disc buffer
BS.1:	DATA Zeilen
BS.0:	M-Lader mit COPY-Befehl

PIN-BELEGUNG FÜR ZWEI JOYSTICKS QUICK-SHOT II, OHNE BEEINFLUSSUNG

Die 5 Dioden dienen der Entkopplung beider Joysticks. Sie unterscheiden sich nur im gemeinsamen Rückleiter. Die vorhandene Elektronik für die "Auto-Fire-Option" sollte man am besten ganz herausnehmen. Da gibt es übrigens verschiedene Varianten: Version 1.3 hatte z.B. einen Multivibrator aus einem NE555, Version 2.0 von 1985 nur noch zwei Transistoren. Der übrige Leiterbahnverlauf ist aber weitgehend gleich geblieben. Zur Unterbringung der Dioden genügt es, kurz vor den Steck-Kontakten die betreffenden Leiterbahnen durchzutrennen und die Dioden aufzulöten. Polung beachten! Den überflüssigen Schalter für "Auto-Fire" kann man ja umbelegen, daß er in Stellung "ON" den Fire-Knopf überbrückt. Beim TI-99/4A reicht das für Auto-Fire.

5*1N4148 o.A.

PIN-NUMMER AM 9pol. SUB.MIN.-D-STECKER

	JOYST. 1	JOYST. 2
LINKS	5	5
FIRE	4	4
OBEN	3	3
UNTEN	8	8
RECHTS	7	9
COMMON	7	2

Dipl.-Ing. M. Becker

ANSCHLUSS EINES AKUSTIKKOPPLERS

Mit einem handelsüblichen V.24 Kabel zum Anschluß eines Akustikkopplers ist man normalerweise gut bedient, solange man z.B. den S 21 d Koppler benutzt. Will man aber den HITRANS 300C Akustikkoppler an seinen TI 99 bzw. an die Schnittstelle des TI anschließen, so entstehen mit einem „normalen" Kabel Probleme, denn das Senden von Daten ist damit nicht möglich. Um diesen Mangel zu beheben, muß zum Lötkolben gegriffen werden: An dem Stecker, der rechnerseitig verbunden wird, also nicht mit dem Koppler, müssen zwei Brücken gelötet werden. Die eine muß Pin 4 und Pin 5 verbinden und die andere Pin 20 mit Pin 6. Ist dieses geschehen, steht dem problemlosen Datenaustausch nichts mehr im Wege.

Stecker zum TI für Hitrans 300

1 13

14 25

Hitrans 3 Koppler Brücken Pin 4 u. Pin 20

Axel Ludewigs

BÖRSE

FÜR BASTLER *** FÜR BASTLER
Verkaufe über 40 Seiten TI-Schaltplan zum Preis von nur 10 DM. 10-DM-Schein + 1,30 DM Rückporto an: Manfred Schneider Nunkircher-Str. 42a, 6648 Wadern-Büsch feld, Tel. 06874/6705

Wir bieten Einkaufsrabatte nur für unse re Mitglieder bei TI99 Hard- und Soft ware an. Das sind Preise zum in die Luft gehen. Info 50 Pfg oder Clubheft DM 3,- anfordern. - TI-Club Baunatal, Matthias Orf, Birkenallee 34, D-3507 Baunatal 1, Tel. 0561/497990.

Verk. Comodore C16, Datasette, Programme oder tausche geg. TI/99 4A.
S. Mertineit, Tel. 0651/67479.

Suche Handbuch zur DFÜ. und andere Bücher. Biete Progr. für E/A und EXB.
P. Mertineit, Tel. 06588/466.

Wir suchen TI-User zwecks Programmaustausch.- Klaus Kinslechner, Eibesbrunnergasse 1/10, A-1120 Wien.

Tausche TI-99/4A Konsole + Netzteil + Kassettensoftw. gegen min. 2 der folg. Module: Statistik, Datenverw.,Terminal Emulator II, Pers. Record keep., Musik Maker, Pole Pos. Möglichst mit Anleitung! - Andreas Scholz, Londoner Ring 6, 6700 Ludwigshafen, Tel. 0621/666424.

Verkaufe über 300 Superprogr. für nur
DM 150- incl.Cass.kosten. Listen anfordern bei: Manfred Schneider, Nunkircher Str. 42a, 6648 Wadern-Büschfeld/Saar, Tel. 06874/6705.

TI-Kass., jeweils DM 10,-:
TI-Grundkurs, Finanzberater, Marketing-Planspiele, CK-TI1A.
TI-Module, jeweils DM 25,-:
Hustle, Amazeing, Hangman, Connect four Kniffel, Othello. - Peter Hielscher, Am Wall 22, 4401 Saerbeck.

Suche Extendes-Basic mit dt. Handbuch, Rec. Kabel. Michael Klingshirn, Uhlandstr. 2, 8933 Untermeitingen

Verkaufe div. Org.-Module (z.B. E/A) und Software (Disk. z.B. TI-Artist). Keine Raubkopien. Liste gegen RP. Evtl. auch Tausch. Martin Voß, Timm-Kröger-Str. 3, 2000 Schenefeld

KOBOOMMM
Verkaufe Zweit-Ex-Basic Modul! Original TI! Mit dt. Handbuch! V.B. DM 130,-!
Wo? Info bei: Manfred Lipwski, In der Wanne 165, 4620 Castrop-Rauxel 4,
Tel. 02305/72237.

Suche RTTY Fernschreiber-progr. für TI99 + Anleitung für TE II. Suche Kontakt zu TI-Usern Raum Kirchheim-Teck-Lenningen. Tel. 07026/5885 - Biete mehrere Schaltungen für RTTY-Converter!

512 KByte Epromkarte für Peri Box ab
280 DM!! - 32 KByte für Peri Box 175 DM, 64 KByte Gramm Modul 288 DM, Modulgehäu se 8 DM, Modulgehäuse 10 Stck 50 DM, Othello 15 DM, Statistics German 20 DM, Parsec 30 DM, TI-Invaders 25 DM. Info anfordern bei:Rainer Moor, Gerstenkam 1 4700 Hamm 4, Tel. 02381/789030.

Haben Sie Probleme mit Ihrem TI99/4A?
Wir probieren, Ihnen zu helfen. Rufen Sie uns an beim TI-Club Baunatal unter 0561/497990! Jederzeit erreichbar.

Wer hat Vulkanien? - Erwin Kinslechner, Eibesbrunnergasse 1/10, A-1120 Wien.
Meldet euch bitte !

TI-Bücher: Jeweils 10 DM.
Tips & Tricks, Spielen + Lernen + Arbeiten, Programm für den TI99/4A, TI-Kochbuch I+II, Spielprogr. selbsterst.I, II.
P.Hielscher, Am Wall 22, 4401 Saerbeck.

Kaufe Sprachsynth. Ang. an: A. Wyrobek,. Hassellstr.116, 5650 Solingen.

KABOOMMM
Verkaufe: Adventure Modul + 11 Adventure Cass.! Alles Orig. TI! V.B. 140 DM & Wo? Info bei M. Lipowski, In der Wanne 165, 4620 Castrop-Rauxel 4, tel. 02305/72237.

32-KB-Speichererweiterung mit Centronic-Schnittstelle (ext. erw.) mit Druckerkabel für 200 DM zu verkaufen.
P. Hilscher, Am Wall 22, 4401 Saebeck.

Verkaufe Othello, Yathzee, Munchman je nur 20 DM. Alle 3 zus. nur 50 DM incl. Porto & Verp. - Mathias Orf, Birkenallee 34, 3507 Baunatal 1, 0561/497990.

We are looking for TI-Fans all over the world! E. Kinslechner, Eibesbrunnergasse 1/10, A-1120 Wien.

TI99/4A + Sprachsyntesizer + Ext. Basic + Spiele + Bücher VB 350 DM.
Roland Albus, Euler-Chelpin-Str. 34,
8900 August, Tel. 0821/793265

Programmiere Ihre Eproms:: z.B. für Modulplatine, Q-Save-Modul. Suche günstig: Sprachsynthesizer, Modulgehäuse, defekte Hardware (z.B.Konsole bis 30 DM Ang. + Info: Peter Rieger, Talstr. 64, 6750 Kaiserlautern, Tel. 0631/73669

Tausche TI-Invader + TOMB-STONE CITY geg. Schachmeister o.Schachmodul für TI99/4A. Thomas Kammel, Kurt-Schumacher-Str. 29, 6700 Ludwigshafen, 0621/655367 ab 20 h.

Verkaufe TI-Basic/XBASIC dt. 35 DM;
Assembler-Kurs-ASEM4, incl. Disk. 72 DM;
XBASIC-Utilities (Befehlserw.) 75 DM.
Werner Zacher - Tel. 089/1572559

Suche: Gut erhaltene TI-Konsole (voll funktionsf.) Preis: VHB.
Tel. 06721/41804 Di - Do 18 - 20 Uhr

Verk. Module! z.B. Minimen + Spez. I
125 DM, Amazing 20 DM; Attack 15 DM usw. Helmut Reiter, Peter-Böhler-Str. 12,
6000 Frankfurt/M 50, Tel. 069/521601

Wer tauscht mit uns Progr. in TI + XBas.
ca. 140 Progr. (Spiele + Anw.)
Liste an: Holger Berneth, Bayreuther Str. 24,
8581 Heinersreuth, Tel. 0921/43462

32K-RAM für 130 DM. Wird seitlich angesteckt! Gr. Vorrat! Per Nachn. bei:
C. Mohr, Friesenweg 38, T. 02845/21507

Verk. TI-99/4A, Spiele, Listings und Module zu Minipreisen. Info gratis von: Markus Theimer, Schwedenstr. 2, 6203 Hochheim

TI99/4A mit Erw. Box RS232, 32K, 2 Lw., alle Module nur kompl. Tel. 0551/22617 Verkauf - Verkauf

Verkaufe TI-99/4A + X-Basic + Datasette + Rec. Kabel + TI-Grafiktablett + Popeye + MxAxSxH + DI4 DVG + Defender + Burgertime u. a. Module. Auch einzeln.
Preis auf Anf. billig: Tel. 0711/766586

Verk. Orig. Rec. Keeping Modul (60 DM). Rec. Kabel 2fach (20 DM). Alle Artikel sind neu und in Orig.Verp. Außerdem suche ich Leute, die mit mir XB-Progr. tauschen. Schreibt an: Detlef H. Barth, Herkulesstr. 43, 5000 Köln 30

Verkaufe P-Code-Karte mit UCSD-Pascal-System incl. Austauschprogr. f. bel. UCSD-Computer; Laufwerk extern TI; Modul Statistik; Invaders,.. Car-Wars, Video-Games II; C-Kab. Leischner, Oberkirch/Baden, Tel. 07802/1464

Suche EX-BASIC mit deutschem Handbuch + Programme. EKB bis 80 DM. Kienle, Maienstr. 6, 8901 Gablingen

Tausche: Orig.TI-Modul Attack + 30 Spiele in TI-Basic gegen: Miner 2049, Indoor Soccer oder Poke Position. Rolf Crisovan Krebsriedgasse 16, CH-6460 Altdorf

Hilfe! Hilfe! Hilfe!
Suche dringend Hardcopy für den EPSON FX-80. Habe XBasic + 32K + Centronics. Zahle fast jeden Preis!!! Bitte meldet Euch unter Tel. 04731/1237

Verk. TI99/4A m. Cass. Rec. + Kabel, Ext.Basic + 128K RAM Speichererw. Beides erst 7 Wo alt. Joyst., Lektüre + Handb. Parsec, Soccer, Tomb. City u. ca. 40 Progr. Preis DM 1300,-- VB, evtl. einz.
Tel. 05331/43459, A. Biedermann

Suche TI99/4A Schaltpläne. Pinbelegungen von Modulport u. i/o. Bus. Adressbele gungen, Systemorganisation. Schaltpläne von Peripheriegeräten sind auch willkommen. Preis VS Tel. 02353/5834
ab 17.oo Uhr

TI99/4A Konsole mit Zubehör Ext.Basic, MiniMem, 3fach Exp. Peri-Box, 32K-RAM, 2 eingeb. Disks m. Contr. (DS/SD) MBI TI auf Centronics (durchgef. Bus), div. Bücher zu verk. Ges. VB 850 DM.
Tel. 08131/93046 nach 18.oo Uhr

BÖRSE

Verk.: TI99/4A, 2 Rec. Kabel, Rec., ca. 40 Spiele + Progr. Sehr viele Listings, Joystick, Bücher: TI Progr. (Spiele, Grafik, usw.). Tips + Tricks, Spielen/ Lernen/Arbeiten, Handb. Writing Work book, TIBasic/ Ex.Basic f. Anfänger.
Tel. 05257/2776 (Joachim Hubertus anfr.)

Verkaufe TI99/4A mit den Modulen Ext.-Bas., Datenvv. + Anal., Stat., Schach, Parsec, Munch Man, TI-Inv., Chish. Trail, Joystick m. Adapter, 25 Zeitschr. 3 Bücher, 350 Progr., Cass. Rec. m. Kabel zus. 400 DM.T.02821/48993

Verk.: Disk-Contr. 249 DM; Disk-Man. 50 DM; Joyst. 30 DM; XB 175 DM, Attack, Invaders, Chis. Trail je 25 DM; Soccer, Parsec je 30 DM; Con. Four 20 DM; Beg. Gram. 15 DM; Finanzber. 15 DM; Basicprogramm-Rout. 25 DM. Te: 0911/498713, 18h

---

Verkaufe TI-99/4A + XBasic + Miniass. + Assemblerbuch + Rec. Kabel + Monitor, alles für 490 DM. Tel. 02104/46184

Tausche TI-Revue Jahrg. 1985 (6 Hefte) gegen Sotware. Ang. bitte an Peter Voigt
Tel. 02162/56520 ab. 19.oo Uhr

TI99/4A Konsole m. eingeb. 32K-CMOS Erw. zu verk. für 250 DM. Rechner ist z.B. in Ass. bis zu 2 x so schnell, dß diese Erw. wie die CPU auch einen echten 16Bit Datenbus besitzt. Tel. 06257/83247

Suche gegen Bezahlung: Bedienungsanleitung f. TI (auch Fotokopie) + Progr. Bildschirmausdruck (Hardcopy) f. Drucker Kaitec 100 Ex. Suche außerdem: Peribox m. 2 Disclw. + Control + RS232 Schnitt stellenkarte. Tel. 07026/5885

** Hallo Assembler Freaks **
Ich bin neu auf diesem Gebiet und möchte deshalb Ass.Spiele tauchen. Wer ist daran auch interessiert? Bitte melden bei: Michael Best, Lessingstr. 34, 6473 Gedern 1, Tel. 06045/1279

TI99/4A: Floppy (TI) aus Box mit Unterlagen 200 DM, Matrixdrucker Seikosho GP 100 A 240 DM. Suche gegen Bezahlung Bücher von TI-Logo II.1 TI-Forth. Multiplan, Artist (auch Kopien). Suche neuw. Drucker Epson FX-RX 80 oder Star. Tel. 06182/26186

Suche für P-Pox:
Modulgehäuse oder Defekte Steckmodule.
R. Breyer, Alte Schulstr. 14, 4242 Rees 1, Tel. 02851/2711

Wegen Systemwechsel: Verkaufe TI-Konsole 150 DM; Peri-Box 200 DM; Laufw. + Contr. 470 DM, 32K RAM 200 DM; RS232 = 200 DM; Speech S. 80 DM; XBasic 140 DM; Parsec + Alp. 50 DM; div. Bücher 2US. 80 DM, Cass.-Kabel 10 DM; Joysticks 20 DM. Top-Zustand. Tel. 040/6560989

Suche: 2 Disk-Laufw. Speech-Synthesizer E/A-Modul, P-Code-Karte, TI-Multiplan sowie Erweiterungen u.a. für TI99/ 4A. Ang. an: H. Hölzer, Mozartstr. 5,
7514 Eggenstein-Leo.

*** Achtung***
Drucke Ihre Listing: 1 = 1 DM; 3 = 2 DM; 6 = 4 DM. Jedes weitere + 1 DM + 80 PF Porto. Knut Meissner, Marsbergerstr. 25, 3549 Diemelstadt 3

Munchman (15) Statistik (15) DV (15) FIN (10) Wumpus (10) Beg. Gr. (10) Video G (15) Parsec (15) L + Est (15) Investre (10). Bastert, Tulpenweg 61, 469 Herne 2

Verkaufe: TI (neu) m. Handbuch --- Suche Ext.Basic II/ Plus!!! Geb. an: E.D.Donig, Frankfurter Str. 15, 3590 Bad Wildungen, Tel. 05621/4298

Hallo TI-User im Großraum Stuttgart + Pforzheim!! ** TI-Club Errorfree **
Sucht Mitglieder!! Info Tel. 07042/24415 oder 77629

*** Achtung***
Schreibe Ihr individuelles Datenverwaltungsprogramm. Das kostet Sie nur 10 DM, eine Beschreibung Ihres Wunsches und Ihre Telefon-Nummer. Knut Meissner, Marsbergerstr. 25, 3549 Diemelstadt 3

32k-CMOS-Erweiterung (superschnell, dß mit echtem 16 Bit Datenbus) zum Einbau in die Konsole. Preis 120 DM. Kompl.Lötarbeit erforderl. Verk. auch Buch 99 Spez. 1, Preis 20 DM VB. T. 06257/ 83247

Verkaufe: Number Magic, Demolition Devision, Dragon. Mix, Alien Addition, Me te or Multiplikation, Alligator Mix je 10 DM; dt. Handbuch zum Extended Basic 25 DM. Tel. 0911/498713 ab 18.oo Uhr

---

Verkaufe: 32 K Intern Origal TI 200 DM.
Module: Ext. Basic (TI) 100 DM; Mini Memory 100 DM, TI-Writer 200 DM; Schachmeister 40 DM; Buchungsjournal (o.Anl.) 60 DM; Pers. Rec. Keeping + Pers. Rep. Gen. 80 DM; Disk Manager 2 DM 40, Par
sec 30 DM, Invarders 30 DM. Bücher: 99 Spez. 1 - 20 DM; TI Basic/Exbasic für A/F 15 DM; Prog. f. den TI (Heigerm.) 15 DM.- Erich Niedergriese, Ronenselstr.9 5800 Hagen 7, Tel. 02331/402220. PS. Gibts in Hagen keine TI/User? Meld.Euch.

Verkaufe: TI-Forth. Multiplan, Disk-Fixer von Nauarone. Alles in einem Mudul zu 350 DM. Tel. 02571/2411

Verkaufe batteriegepufferte 32K Speichererweiterung zum Anschluß seitlich an die Konsole, mit durchgeführtem Bus und LED zur Betriebsanzeige, für 180 DM. Bei Bedarf dazu passendes Assemblersoftware (Ramdisk,...). 1 orig. Extended Basic Modul für 190 DM (alles neuwertig).
Gregor Lohmann, Bleibergerstr. 54, 5100 Aachen, Tel. 0241/86447

P-Box + Disk. + Contr. + Man. + 32K zus. 1250 DM; TI99/4A + Ex-Basic 300 DM; Edit Ass. + Handb. in dt . + engl. + Kurs 11 320 DM; Sprachsys. + Modul 130 DM; Eprom + Progr. 130 DM; Ex-Basic Handb. in dt. + engl. 40 DM; Spez. I 35 DM; Parsec 45 DM; Mu-Man + Tomc 20 DM. Tel.0203/478723

Zu Verkaufen: 1 TI-99/4A m. Org.Box + Laufw. + 32K + Contr. 1300 DM; Multipl. 240 DM; TI-Writer 260 DM; Datenverw. + Analy. + Pers.-Rep.-Gen. je 65 DM; Haushalt-Mod. 90 DM; FiHn-Mod. 60 DM; od. kompl. 1800 DM, Ext. 32K 180 DM.
Tel. 069/414539, evtl. m. e. Progr.

32KB-Speichererw. f. TI99/4A f. Peribox sowie Ext-Basic-Modul m.Buch (dt) 270 DM VB. Erwin Wachter, Gerstbuschstr. 9, 6670 St. Ingbert, Tel. 06894/ 6890

Verk. TI99/4A + Ext.Basic + Rec. + Kabel + Joyst. + 4 Fachbücher + Basickurs +
100 Progr. auf Kass. f. 450 DM. B. Dohr, Tel. 02234/81364

Verkaufe: Videogames 1 + Attack (55 DM), Diskmanager 2 (40 DM), TI-Joysticks (40 DM).- Kaufe Schachmodul (45 DM) oder tausche gegen Videogames 1 + Attack. Verkauf per NN. Kai Breuer, Zur Burg 30, 6393 Wehrheim 1

Verk. TI-99/4A (80 DM), Ex-Basic (110 DM), RS232 + V24-Kabel (160), Speech-Syntehsizer (100 DM VB), Rec.-Kabel (10 DM), Joyst.-Adapt. (10 DM), 2 Bücher (35 DM), Munchman (29 DM), alles ohne Defekt
Tel. 06074/29556, bitte Daniel verlangen

Zu Verk.: 2 x P-Box, 2x RS232, 3x 32K, 4x Joyst., 4x Rec.-Kabel, 1x Sprachsynth 1400 Progr./180 Disks, Ex-B, E/A, Mini-Mem, Modem, TI-Writer, viele Module, viele Bücher, viele Extras. Tel. 06204/ 1471 (Piepton abwarten, Name, Nr. und Ang. geben.

TI99/4A Verkaufe:
Konsole/Box mit Disk. + Contr./interne 32K/ext. Druckerinterf. + Druckerkabel, Ex-Basic org. sowie div. Module + Bücher
Tel. 069/7438368, Schmidt, v. 9 - 16.oo, ab 18.oo Uhr Tel. 06035/3214

Verkaufe P-Box mit Disk-contr., 1 Laufw. 5 1/4" für 900 DM, Atronic RS232 Karte mit 32KB 7 Mon. alt für 370 DM, TI-Writermodul für 200 DM. Holger Wesemann,Kkreuzkrug 62,
3079 Raddestorf 1

*** Finito Aus Schluß Vorbei ***
An alle meine Tauschpartner: Ich verkau fe meinen TI, bis ich ihn verkauft habe wie folgt: TI99/4A + XB 2 + 32K + 3Mod. + Joy + 250 Pgm + Lit. + Rec. (nicht Orig.) NP 1250 DM - VB ca. 800 DM.
Michael Bolte, Am Sportplatz, 3065 Nienstädt, Tel. 05721/ 2627

Achtung! Biete folgende: 1 MiniMemory 110 DMd; 1 Musicmaker 40 DM; 1 Exb-Lern auf C. 15 DM; alles zus. 145 DM. Thomas Pister-Talstr. 49, 8770 Lohr
Tel. 09352/3804 (ab 20.00 Uhr)

Suche Spiel u. andere Module sowie Zubehör für TI. Ang. an: S. Engelbert, Heinr.-Schütz-Weg 5, 7151 Allmenbach oder anr.: 07191/58233 abends

BÖRSE

TI-User möchte Enterpr. + B. Rogers nicht nur spielen, sondern sehen! Wer kann mir in V2000 Folgen von Sat1 aufnehmen? Zahle Cass + Bonus, nur schrift. Anfr./Ang. an: P. Schmitz, Am Zehnthof 4, 5480 Remagen 4

TI99/4A Box, RS232, 32K, Lw., XB, TI-Writer, Statistik, Pers. Kep., Pers. Gen., Disk-Manager, viel Softw., evtl. auch Drucker VP: Gegen Höchstgebot!
Tägl. ab 18.oo Uhr, Tel. 09123/ 12249

TI99/4A Ext-B 32K, TI-Spiele 1+2, Bücher Cass./Module/ Schaltpläne/Rec./Zubehör.
VB 510,--, Tel. 0231/391541 ab 20.oo Uhr

TI99/4A + Rec. + Module: Schach, Othello
Datenverw. + Lit. 499 DM.
Tel. 02247/2611, R. Lehmann

Verkaufe TI-System:
Konsole 110 DM; Ex.Basic 140 DM; Minimem 130 DM; E/A m.dt.Handb. 170 DM; Box, Contr., Disk nur kompl. 650 DM; 32K-Karte 200 DM; RS232/Centr.-K 200 DM.
C. Hasselkuss, Zaunkönigweg 5, 4630 Bochum, Tel. 0234/ 311411

Suche Ass.Progr. f. E/A + P-Box! Auch Programmtausch! Suche weiterhin die Anleitungen von S.X.B., Protector II + FIG-FORTH (Weiand)! Verk. ATARI-Modul "Moon Patrol" f. .50 DM! M. Rohrmoser, Nietzscheweg 16, 4300 Essen, 0201/530687

Verkaufe TI 99/4A + 32K Erw. + Centr. + Kabel + Ex-Basic + Buchungsj. + Meteor Multipl. + Yathzee + 130 Spiele +TI-Finanzberater + Voodoo Castle + 15 Anwenderprogr. für VB ‹ 800,-- DM. Tel. 0203/332920

Systemerweiterung mit Progr. Speedtape 2000 für M.M. + 32 K, 6mal schneller als orig. auf die Kass. Ass.soft. bis 32K, ohne Disc. + Basicroutinen. Besser komparabel, höhere Datensicherh. als orig. öS 350,- an: Rudolf Mauk, Justgasse 29/67/ 1, A-1210 Wien

---

Achtung!!! Verk. TI99/4A + Ext.Basic + Rec. Kabel + Org. Joysticks + 3 Spielmodule + Lit. für nur 450 DM.
Stefan Lehnert, Georg-Böhm-Str. 16,
8700 Würzburg

Verk. TI99/4A + Ext.Basic + Rec. + Kabel + 21 Spiele + Handb. + Computerzeitschr. f. 400 DM, Tel. 0441/601730

Ex-Box + Disk-Cont. + Laufw. int. + Laufw. Ext. + Disk-Manager 2 + RS232 Karte + Kabel RS232 -Centr. + Othello + Demolition Div. + Alien Add. + Yatzee für 2000 DM. Ext.Basic 120 DM, 32K-RAM Karte Php1260 DM 250.-. Samstag 17-18 h
Tel. 05603/2374

Verkaufe meine voll funktionsf. Zweitkonsole + Netzteil für 60 DM. Verkaufe auch Progr. nur 10 DM. Bitte melden bei: A. Scholz, Londoner Ring 6, 6700 Ludwigshaven, Tel. 0621/666424

Einmalig Verkaufe TI99 Adventure mit Text + Grafik für 15 DM incl. Progr-.Disk zuzügl. Porto. Benötigt X-Basic, 32K Erw., DSK, Lw. Tel. 02621/50713

---

* TI99/4A * C128 * Texas Spiele-Module * zu verk. bis zu 60% preiswerter. C128-Malprogr. im 128 Modus in Maschinenspr. Einbinden der Bilder in Basic 7.0 und 2.0 mögl. Tel. 0208/23299 ab 18.oo Uhr

---

Verk. Centronics-Interface ext. 100 DM.
Suche Tauschpartner f. TI-Softw. in X-Basic, Ass. o. Forth. Tel. 04331/31074, ab 16.oo Uhr

---

!! 600,-- DM = TI99/4A + Exbasic + RS 232 + 2 Joysticks + Buchungsmodul + Personaelmodul + Datenrec. + Spiele, Lit.
Tel. 05123/678
Verkaufe aus meiner Programmsammlung To Progr. Eine C-60 Kass. mit ca. 27 Progr. f. nur 10 DM. Bei Interesse melden bei:
A. Scholz, Londoner Ring 6, 6700 Lud
wigshafen, Tel. 0621/666424

TI-Bücher billig! 99 Spez. 1 DM 20; 99 Spez. 2 DM 20,-; Basic/Ext.Bas. DM 20,-; Spielen, Lernen, Arbeiten DM 15,-; Tips & Tricks 20 DM; TI99/ 4A Intern (neu)
DM 18,-. Tel. 040/6560989

Verkaufe meinen Reserve TI99/4A m. RS232 und Brother EP 22, incl. allen erforderlichen Kabeln. Preisidee: 800,- oder GROM-Karte! R. Breyer, Alte Schulstr.14,
4242 Rees 1, Tel. 02851/2711

Verkaufe extern: 32K + RS232; intern: 32K + Contr. + Laufw. Tausche Ass.Progr. Verkaufe Module: Soccer, Parsec, Othello Fitness Training, Text + Datei, DV + Analyse, Buch.-Journal, Early Reading. VHS. A. Goering, Tel. 07243/ 78582

Verkaufe Superspiele in TI-Basic. Olympiß 1984, Dragon-Castle, P.-Progr. I + II, Little Computer People, Miner, Schatzsuche, Flippersim., Rambo II, Formel I, Commander u.v.a.: Liste anf. würde auch tauschen. Rolf Crisovan, Krebsriedgasse 16, CH-6460 Altdorf

---

Verkaufe: TI99/4A 110 DM; XBasic 140 DM;
Minimem 130 DM; E/A m.dt. Handb. 170 DM; Box, Contr., Disk 650 DM; 32K 200 DM; RS232 200 DM; Schach 50 DM; Parsec, Othello 50 DM; Joyst. 35 DM; Bücher, Zeitschr. 100 DM... C. Hasselkuss, Zaunkönigweg 5, 4630 Bochum, 0234/ 311411

Suche TI-User. Zwecks Softw.-Austausch. Nur Kass. an: Marco Göbel, Ernst-Reuter-Str. 9, 3550 Marburg

---

Suche Diskette TI-Artist Version 2.0.Verkaufe MBi-Interface. TI99/4A AWF Centronics. Preis 140 DM. T. 06196/42411

Verk. TI99/4A + Ex.Basic + TI Invaders + Cass.Rec.Kabel + VTel Literatur + Joyst. Adapterkabel f. 260 DM. Auch einzeln.
Dieter Schafhuber, A-6200 Hinterriß 7

Verk. TI99/4A, P-Box, 32K, Laufw., Contr., RS232 m. Kabel, X-Basic, E/A, Writer, Schach, Advent., Invader, Munchman, K-Rek., div. Pr. u. Spiele auf Disk u. Kas., Zeitschr. u. Bücher VB 1200,-, auch einz. Tel. 0431/563724

TI-Computer Benutzer Club Schweiz, Postfach 322, CH-4528 Zuchwil - Steckbrief: Mitgl. 300 TI99/4A Besitzer, unterteilt in 5 Sektionen. Du erhälst weitere Infos! Fordere sie an mit einer Postkarte! Gruß Euer TILI

Verkaufe TI99/4A Versch. Module, 10 Org. Dynamics Cass., 13 Org. Cass. m. Spiele Speech Sunth. ExBasic. Viel Lit. u. Zeitschr., Org. Sticks u. Adapter f. ATARI Sticks 680 DM, Tel.08122/20624 18h

---

HELPHELP***
Du hast mir auf Grund meiner Anzeige die ADV-Lösungen geschickt. Als ich Dir nun die Disk. zurückschicken wollte, fand ich Deine Adr. nicht mehr! Bitte melde Dich bei mir. C. Jarque, Ludwigsburger-Str. 14, 7257 Ditzingen 1, T.07156/34941

ACHTUNG
Suche: Diskettenlaufw. bis 350 DM (ext.) + 32K-Erweiterung mit Centronicsschnitt
stelle für bis 200 DM. Knut Meissner,
Marsbergerstr. 2, 3549 Diemelstadt 3

HALLO! Noch Zubehör gefällig? Leicht def. TI mit 32K int., Exb., Speech-Synt., Progr.-Rec. + Kabel, Joyst., Addap., Quicksh. Joyball, Module: z.B. Pole Posit., Schachm., Hopper, 10 Ass. Progr. auf Kass., Software, TI-Rev., Bücher. Auch einzeln! 32K ist übrigens noch voll in Ordn.! Preise ist VHB aber günstig! Michael, Tel. 06127/5652

Texas Instrumens-Fans
- TI-Rev. von der 1. Ausg. - einschl. Heft 4/86 + Sonderh. 1/85 + 2/85 + jede Menge Listings aus and. Heften kmpl. nur 50 DM.
Tel. 04191/60215

Verk. wenig gebraucht: Matrixdrucker Seikosha GP 100 A, 80 Zeichen/Zeile m. Hardcopy-Progr. f. 250,— DM. N. Burkhard, Ringstr. 21, 6530 Bingen 17.

Zu verk. Handb. z. Advertizer Modul 20,— Fr. Brief-Vorauskasse o. NN. B. Schönauer, Alpenstr. 32, CH-2540 Grenchen.
Suche Eprombrenner + Software

Diskettenlaufwerk Sliml. 5–1/4"DS TEAC FD55B 160,—
Einbausatz 2 Laufwerke + Stecker + Zeichnungen für TI 99/4A Box komplett 320,—
GROM-Platine + Gehäuse 22,—
Module Statistics 20,—
Module Othello 20,—
E.L.Levits
Roggenkamp 3
2000 Hamburg 60
040/2994609 wegen Urlaub ab 1.9. nur schriftlich